SKEYS
BY
INOVA

INOVA
11311 STEMMONS FRWY, SUITE 7, DALLAS, TEXAS 75229

SKEYS - for the Osborne Executive(tm)

Version 1.1 - July, 1985

by Martin Murray

SKEYS gives the Executive owner over two dozen additional functions, which take up no program space and are available to the user even in the middle of most other application programs. These functions include a calculator with two memories, a function-key editor, "smart" function keys (that can call other function keys or prompt the user for input), four extra sets of function keys, a redefinable keyboard, 22 extra function keys (assignable to just about any key, with up to 255 characters each), an ASCII chart, a screen-print, typewriter emulation, on-line quick reference guide, instant arrow key redefinition, time/date/last calculator answer on special function keys, printer control, clear screen, PROGRAMMABILITY, and tools for the programmer, like a hexadecimal 16-bit calculator and a memory editor, plus much, much more.

****

The folks at INOVA prefer working with microcomputers over almost anything else. We believe in producing quality programs, as evidenced by the programs we have already written for the Osborne 1 and Executive, and others, and released for non-commercial distribution. We think there are still unaddressed software needs among owners of CP/M microcomputers, and want to address those needs.

In order to continue our programming efforts, we really need your support - both as to any types of programs you wish considered, and adequate yet reasonable reward for our efforts. Please encourage your friends and associates to purchase a copy of our program for themselves, if they want it, rather than your giving or selling copies of this program to them.

****

SKEYS(tm) -- For the Osborne Executive(tm)

PROLOGUE

Long, long ago (it seems), Osborne Computer Corporation set out to build the best microcomputer of its time, using the Osborne 1 as a springboard and listening to the feedback generated by the users of that landmark machine. The word "best" is a qualitative word, something like "beauty" in that it is often only in the eye of the beholder. Yet, OCC's engineers actually designed a machine with an incredible amount of potential -- potential to become what would be to the average computer what the average computer was to nothing.

That potential was not realized. The marketplace never even knew about, much less appreciated, the capabilities of the Executive, choosing instead to listen to the hype of the pseudo-16-bit machine of the time. (It didn't matter that you could "do it faster" on an Osborne -- everyone seemed to know that "sixteen bits was better than eight".) OCC ran into hard times shortly after introducing the Executive, and not many were sold.

Yet those machines that were sold stuck stubbornly around. Even with the unavailability of repair (for a short while), many of you stayed with your Osborne.

That pseudo-16-bit market is where the money is, but there are a couple of us who have grown to appreciate the power of the Executive. Unlike most 8-bit machines, the Executive was designed to use more than just 64K bytes of memory. In a way, it is very similar to the pseudo-16-bit machines which can also address more than 64K of memory -- they, too, have to use a "segmented" architecture, except theirs is built into the processor and the Executive uses additional circuitry to accomplish the same purpose.

Well, those of us at INOVA know that we can't make much supporting an 8-bit machine like the Executive. Although we would like to make money, we'd also like to design software for good machines. The Executive is a GOOD machine. That's why we wrote SKEYS. It's designed for those who already know how to use their machine and want to get even more out of it. We believe SKEYS shows you some of the potential mentioned above, and will make your computing life easier and more enjoyable.

If you have any constructive criticism after using SKEYS, please let us know, that we might consider it for future software development. If you have something nice to say, let us (and others) hear that, too. It's almost as good as money.

TABLE OF CONTENTS

PROLOGUE 2

INTRODUCTION TO SKEYS 4

INSTALLATION 5

NOTES 7

FUNCTION OVERVIEW 11

SKEYS FUNCTIONS 14

- Quick Arrow-Key Redefinition 14
- The On-Line ASCII Chart 15
- The Ten-Key Business Calculator 16
- The Programmable Calculator 22
- Get String from Video Display 25
- Examine/Edit/Redefine Your Function Keys 26
- Comments in Function Keys 28
- The Hexadecimal/Binary/Octal/Decimal Calculator 29
- The Interactive Function Key 31
- Exchanging Function Keys Sets 34
- The Memory Editor 35
- The Control-Shift-W Command 37
- Printer Control 40
- Print Your Screen 43
- The Skeys of SKEYS 44
- The Typewriter Emulation Mode 47
- Key Redefinition 48
- Miscellaneous SKEYS Functions 49
- SKEYS Within SKEYS 52

SKUTIL, THE SKEYS UTILITY 53

OTHER SUPPORT PROGRAMS 55

- COPYSKEY, FKEY, SKEY, WSPATCH 55
- The DVORAK Keyboard Files 56

APPENDIX A -- Restarting 57

APPENDIX B -- Resistor Fix for Contrast 59

APPENDIX C -- Application Notes 60

APPENDIX D -- SKEYS Technical Notes 65

UPDATES, LICENSES, WARRANTIES 69

INDEX . . . . . The Last Page

INTRODUCTION

SKEYS, a program laboriously written by Martin Murray (the author of NULU, et alia) contains over two dozen popular and powerful functions to enhance the use of the Executive computer. The program is unusual in that it gives you those functions in what we call "background" mode, i.e., once SKEYS is run, it sits in your computer, waiting in the background, while you run other programs, or even change diskettes. Then, when you need a function, it's there, only a keystroke away, even in the middle of such programs as WordStar(tm), Personal Pearl(tm), MBASIC(tm), SuperCalc(tm), and many others. SKEYS even has a built-in list of the functions available, a "help screen" that pops up any time you press a control-shift-? key combination. And it does all this without even using any of your program space, because it resides in a different bank of memory.

We ask you for your patience in learning the functions of SKEYS (pronounced "skis") available to you. This is because there are quite a few functions, and some of them are very powerful, requiring a good understanding of the Executive in order to make the most use of those functions (e.g., the memory editor). However, you do not have to know all the functions to use and enjoy SKEYS -- if you accidentally get into a function you did not wish to use, there is an easy way to get out.

First, please read the INSTALLATION and NOTES sections following this introduction. We then suggest that you read through the brief overview of the functions that are available to you, and choose those functions that you would find immediately useful (like the calculator, the function-key editor, and the screen-print feature, for example). Then, turn to the corresponding pages in the manual and read about each function you want to use. Each function's description is meant to be complete in that it does not require the knowledge of other functions to use it.

However, other functions may be referred to if they can contribute to the usefulness of the function in use. For example, you can "print your screen" -- dump what is displayed on your screen to your printer -- with control-shift-P, but before you print it you can edit it by using a different function -- control-shift-G.

After you have become comfortable with these functions, you might want to look at some of the other capabilities of SKEYS. A utility, SKUTIL, is provided to save the extra function key sets and the special "skeys" function keys, as well as any new keyboard you may have created. This utility is discussed in the section following the SKEYS function descriptions.

ACKNOWLEDGEMENTS: INOVA wishes to thank anyway the folks who inundated us with their occasionally constructive suggestions on the programs and manual: Jo Allan, Peter Cole, G. D. Eckler, Doug Griswold, David Krasner, Don Lindsly, Dave Mack, Thomas Shea, Lowell Streiker, Jim Switz, Rick Trow, Gary Verwers.

INSTALLATION

We recommend that you make a backup copy of the supplied diskette, and put the supplied diskette away in a safe place, away from dangers like magnets, motors, and pirates.

****
* *
* **NOTE** *
* *
* If you have a "bank 2" version of SKEYS (one requiring *
* the INOVA 500 memory expansion card), you DO NOT have *
* to install SKEYS and can go on to the next page. *
* *
****

Otherwise, before you can use SKEYS, you must "install" it. This is a simple procedure; just run the PATCHCPM program that we have provided. To do so, first "boot" a diskette containing the CP/M Plus system and get to the A> prompt. (If you have loaded an application like WordStar or SuperCalc, exit from the program as described in the application's documentation.) Put a copy of the SKEYS diskette in the B drive, then type:

B:PATCHCPM<cr> where <cr> means press the RETURN key

You will be prompted for a drive which contains the CPM3.SYS file to be patched. Enter the drive letter, or press ESC to exit. After an appropriate letter is entered, PATCHCPM will look for the file on the specified drive, and proceed to patch it. It will tell you which version of the BIOS (1.0, 1.1, or 1.2) that it is patching, then it will return you to the CP/M prompt. That is, if all goes well. If the diskette does not have the file CPM3.SYS on it, you will be told. Every bootable diskette must be patched if you want SKEYS to run after booting with that diskette.

PATCHCPM bravely goes and tells the CPM3.SYS file not to use a certain portion of the memory in bank 0. This is the portion of memory that SKEYS will use when it is loaded.

NOTE: PATCHCPM is the only sarcastic program we have provided. It's the black sheep of our family, but it does the necessary dirty work, so we put up with its peculiarities. Please bear with us.

After you have patched your operating system, press the RESET key and re-boot. **A PATCHED SYSTEM MUST BE RUNNING** before SKEYS can be executed. As they say, "FAILURE TO COMPLY WITH THIS REQUIREMENT WILL RESULT IN DIRE CONSEQUENCES," consequences like the error message: "How could you? This version of CP/M won't let me run!"

Now, copy SKEYS.COM and SKEYS.OVL to your boot diskette with PIP or other file-copying utility. Then type:

SKEYS<cr>

and SKEYS will display a sign-on message and return you to the CP/M prompt.

If you attempt to load SKEYS after it has already been loaded, SKEYS will recognize the fact, and merely display the signon prompt.

Please note that SKEYS cannot be loaded from the R option of WordStar. SKEYS can only be loaded from the CP/M prompt. Also, if you use a PROFILE.SUB file to automatically load in several programs when you boot, SKEYS must be the last program in the file.

You can automatically load SKEYS with a modified EXECST. We have supplied an excellent EXECST.COM-maker written and released for non-commercial distribution by Peter Cole. It will create a special EXECST which can load SKEYS and then chain to your special application, like WordStar or SuperCalc. The program, MAKEST, has been well received by our beta-test-site users. You can find it with its documentation on the SKEYS diskette. MAKEST is provided here with Mr. Cole's kind permission.

NOTES

(The following notes are good things to know when reading this manual or using SKEYS.)

The first good thing to know about using any SKEYS function is **how to get out** and back into the regular world. No matter where you are within SKEYS, you can always exit, one level at a time, with the **control-shift-ESC** combination keystroke. -- Well... maybe the first good thing to know is what a "control-shift-ESC combination keystroke" is. To get this keystroke, hold down both the control key (marked "CTRL" or "CNTL") and the shift key (right! -- "SHIFT"), then briefly press ESC. Sometimes, if you're not too dyslexic, you can do this with one hand. (Nobody at INOVA can.)

So, if you accidentally call up a SKEYS function and "get lost", just press and hold control-shift-ESC for about 5 seconds. This will return you to your application.

All the SKEYS functions are invoked (called up) by control-shift-"key" sequences, with the "key" usually a letter associated mnemonically with the desired function -- that is, the letter is generally the first letter of one of the words used to describe the function.

You can quickly see what is meant by the previous paragraph by typing control-shift-? after SKEYS is loaded. This will display the on-line "quick reference" that looks like this:

SKEYS Functions -- Press RETURN to exit

Each function is accessed by holding down the control and shift keys, and pressing one of the following keys:

A ASCII chart	V Replay video string (see G)
C Calculator with memory (see =)	W Replay word @ mem-editor cursor
F Function keys examine/edit	X Exchange key definition
G Get string from video display (see V)	Z Zap (clear) screen
H Hex/Bin/Oct/Dec calculator (see =)	= Replay last calculator answer
I Interactive function key begin/end	\ Select background attributes
J Just replay User f-key (see U)	TAB Toggle cursor attributes
K Keyclick toggle	
L Load one of four function-key sets	Define arrow keys as:
M Memory editor	1 CP/M(tm)
O Output char/f-key/SKEY to printer	2 WordStar(tm)
P Print screen (use CTL-SH G to edit screen before print)	3 Special
R Replay interactive function key	4 ASCII null (^@)
S SKEYs examine/edit/redefine	7,8 System time, date
T Typewriter emulation mode	
U User-prompting f-key (RET to end)	ESC Guaranteed exit

Impressive list, isn't it? The author of the program has been known to stare at this help screen for hours, mumbling something like, "I can't believe I wrote the whole thing!"

Some functions do nothing but display useful information, like this help screen and the ASCII chart. To exit from these functions, just hit the RETURN key. (Actually, almost any key will cause an exit, except the CAPS or ALPHA LOCK key, the CTRL key, and the SHIFT keys.)

If you wish, you can "dump" a copy of the screen at just about any time with control-shift-P, so you can actually get a hard copy of the help screen or the ASCII chart. (A printer is required for this option to work well.)

Upbeeps and Downbeeps

If you hear a "toodle" type of beep, you have encountered the "beep" of SKEYS. It simply means that you have accomplished something that was not displayed on the screen, like the switching of your arrow-key definitions (control-shift-1, -2, and -3), or tried to enter a key that is inappropriate while executing a SKEYS function. If you hear a rising or falling string of almost-musical notes, you have begun or ended the interactive function key definition (or forgotten to end it, and the computer ended it for you).

We've attempted to be consistent throughout the varied SKEYS functions. You will be using the same line editor for the function keys, the skeys, and the typewriter; you can print your screen with control-shift-P almost everywhere, to get a copy of your calculations, or your function keys, or the memory editor, etc. Some SKEYS functions can be "called up" while inside other SKEYS functions; a chart is provided to show you which functions can be invoked when.

There Are SKEYS, and Then There Are Skeys

There is a difference between SKEYS the program and skeys the special function keys:

The letters of the latter are lower case. (You already knew that, right?)

SKEYS, the program, is named after the powerful skeys function keys. Unfortunately, we talk about SKEYS functions in this manual, and the skeys function keys, which are but one of the many functions available to you through SKEYS. If we are talking about a "skey", or "skeys", we are referring to those that you define through control-shift-S. If we say "SKEYS", we mean the program SKEYS.

Restarting

It is sometimes possible for you to call up a SKEYS function while your application program is in the middle of another task, and not just merely waiting for you to type another keystroke. In such an instance, SKEYS can suspend the processing of the program which, when returned to, will require "restarting" by the pressing of a key. The key chosen to "restart" the program should be appropriate to that program; e.g., control-C terminates some programs and therefore is usually not a good choice to restart the program.

See Appendix A for more information on restarting.

Contrast

One deficiency on many Executives is that the contrast control is poor. This is due to the fact that the wrong resistor values are used on the main circuit board to give you half-intensity (it's more like one-quarter-intensity).

If your Executive has this problem, you will experience some difficulty reading some of the SKEYS displays. One method of solving the contrast problem is to "invert" the display; see the set-background-attributes function (control-shift-\) in the Miscellaneous SKEYS section.

A more satisfactory solution is to replace the wrong resistors with the right ones. This solution is documented in Appendix B.

The INOVA Font

If you are using the original character font supplied by Osborne Computer Corporation, we highly recommend that you try the INOVA font, included on the distribution diskette (in a file called INOVA.CHR). This font, developed by one of the staff at INOVA, has improved readability on the internal monitor, subsequently reducing eye strain. The font is not copyrighted and may be freely distributed to other Executive owners.

If you do not wish to change your font, skip to the SKEYS editor section that follows.

To install the font for use as the main character set on your computer, use OCC's CHARGEN program as follows:

Put the OCC diskette that has CHARGEN on it in the A drive. Put a copy of the INOVA SKEYS diskette in the B drive. Press RESET if necessary; press RETURN to boot the CP/M operating system. At the CP/M prompt, type CHARGEN followed by a carriage return. The program will display a menu of 7 choices. Select item 2 to read a font from a file. When it asks you to enter the file name, type B:INOVA followed by the RETURN key. After the program has read the file, select item 4 to write the file to the

system tracks, then select the main set at the next question (item 1), finally followed by the letter of the drive containing a COPY of a bootable diskette on which you want to place the new font. If all goes well, then press RESET and reboot to use the new font.

The SKEYS editor

When editing a regular function key, a skey function key, or using the typewriter emulation mode, you can make use of certain editing commands. (A quick-reference line is displayed on the bottom line of your screen during editing.) These commands are:

The **up-arrow key** toggles the insert mode on or off. When insert is on, keys that you press will be inserted before the character at the cursor, and all characters to the right of the cursor will move right -- unless there is no more room for entering characters. When insert is off, characters you type will "overwrite" any existing characters.

The **down-arrow key** deletes the character at the cursor, if there is one, and all characters to the right of the cursor will move left.

The **left-** and **right-arrow** keys will move the cursor left and right through the line.

Control-shift-A moves the cursor to the beginning of the line.

Control-shift-F moves the cursor to the end of the line.

Control-shift-K "kills" or deletes from the cursor to the end of the line.

Control-ESC (or **control-shift-ESC**) aborts the edit, without saving any changes or sending the last line to the printer (if in typewriter emulation mode.)

If you are in the function key or skey editing mode, control-RETURN will save your changes, and allow you to select another key to edit, or exit. If you type a carriage return (i.e., press RETURN) you will enter a carriage return into your function key or skey.

If you are in the typewriter mode, a control-RETURN will put a carriage return (displayed as an underlined M) into your text, without sending anything to the printer. Pressing RETURN, on the other hand, sends the entire line that you have entered to the printer.

FUNCTION OVERVIEW

QUICK ARROW KEY REDEFINITION -- **control-shift-1** defines your arrow keys as CP/M, **control-shift-2** defines them for WordStar, and **control-shift-3** defines them as a special combination of WordStar and CP/M.

THE ON-LINE ASCII CHART -- **control-shift-A** displays an ASCII chart with its decimal and hexadecimal equivalents. Useful for determining printer codes, etc.

BUSINESS CALCULATOR -- so-called to distinguish it from programmer's calculator. **Control-shift-C** or control-shift-B brings up a calculator with two memories. Up, right, and down arrows redefined as +, -, and * for convenience. The slash key (/) is used for division. C clears the calculator. Backslash (\) changes sign of current entry. The return key or equals key completes pending calculation. Memories accessed with M or N prefix followed by +, -, *, /, C (clear), R (recall), or = (store). Decimal point set with F followed by 0 through 9, or F for floating. Last calculation available while in calculator on L key, or on **control-shift-=** key after exiting calculator (via ESC or control-shift-ESC). This means that you can replay the last answer into your application by pressing control-shift-=.

GET STRING FROM VIDEO DISPLAY -- **control-shift-G** allows one to "grab" a string from the video display. Use arrows to move cursor on screen. Control-RETURN causes "capture" of string from cursor to end-of-line, or last non-space character on line. Useful for capturing long CP/M command line for repeat entry. Replay captured string with **control-shift-V**. While in control-shift-G mode, screen can be edited, then printed with control-shift-P. Exit without saving any string via ESC or control-shift-ESC. Control-shift-A and RETURN move cursor to left margin, control-shift-F moves cursor to right margin. Upon exit, original screen is restored.

EXAMINE/EDIT/REDEFINE YOUR FUNCTION KEYS -- **Control-shift-F** displays current set of 10 regular function keys. Use number or letter to select function key to edit or change. (See the information on the skeys editor in the NOTES section.) Function keys can now refer to other function keys, or even themselves. Comments can be entered with **control-shift-"** pairs.

THE HEXADECIMAL/BINARY/OCTAL/DECIMAL CALCULATOR -- This is the programmer's calculator, invoked with **control-shift-H**. Entry of operands can be in several bases. Entry is from left to right. All non-decimal values are suffixed with B for binary, O for octal, or H for hexadecimal. This is a 16-bit, unsigned calculator. Arrows redefined as in business calculator. ESC or control-shift-ESC exits. Last answer on L key while in calculator, and also on control-shift-= key upon exit. Answer saved in hex if control-H is used to exit; octal if control-O; and binary if control-B. Base suffix not appended on control-shift-=.

THE INTERACTIVE FUNCTION KEY -- **control-shift-I** starts and stops the interactive function key. All non-SKEYS keystrokes (up to 255 characters) are remembered while control-shift-I is active. When first invoked, control-shift-I generates an ascending series of notes. When "closed", it generates a descending series of notes. Playback is accomplished through **control-shift-R** (for "replay"). Pressing control-shift-R causes the sequence of keystrokes to be replayed back into the application, or into a function key or skey for later saving to disk.

EXCHANGING FUNCTION KEY SETS -- There are four more sets of regular function keys, in addition to the main set. These are "swapped in" to temporarily replace the main set by typing **control-shift-L** followed by the number of the set. The main set is similarly restored with control-shift-L followed by the letter M. The extra sets are saved with the SKUTIL utility, in program files that are self-loading.

THE MEMORY EDITOR -- Although mainly a programmer's tool, the memory editor can be used to inspect or change the computer's memory directly, allowing the user to eliminate one or more programs from his/her diskette. See the memory editor section for more details.

PRINTER CONTROL -- **control-shift-O** allows the user to output single characters or strings of characters directly to the currently-selected printer. When pressed, control-shift-O prompts for a character, function key, or skey to send to the printer. Useful for sending a formfeed, or setting printer to different print modes, etc., while in the middle of an application.

PRINT YOUR SCREEN -- **Control-shift-P** sends a copy of your current screen image to the printer. Graphics characters are converted into asterisks. Screen image can be edited prior to screen-print through control-shift-G "Get String From Video Display" command (q.v.).

THE SKEYS OF SKEYS -- **control-shift-S** displays the 22 skeys, both the definitions and the keys that the definitions are assigned to. The skeys are similar to function keys, but can be assigned to just about any key on the keyboard, and each can be up to 255 characters long. Skey definitions can refer to the regular function keys. After typing control-shift-S, use the arrow keys to select a skey and enter the definition, or assign a skey to a key on the keyboard. Unassign a skey with control-shift-ESC. Refer to the skeys editor information in the NOTES section.

THE TYPEWRITER EMULATION MODE -- **control-shift-T** or control-shift-E enters the typewriter mode of SKEYS. A one-line editor allows you to edit each line before you press RETURN, at which time the entire line is sent to the printer, after eight spaces are sent for the left margin. (The number of spaces can be changed.) Good for addressing envelopes, or for quick notes,

etc. Since ESC is a character that one might want to send to the printer, you must press control-ESC or control-shift-ESC to exit from the typewriter mode back to your application.

KEY REDEFINITION -- **control-shift-X, key-to-be-defined, new-key-definition** allows you to redefine one key, or the entire keyboard. Keyboards can be saved with the SKUTIL program. Control-shift-X can be used to unassign skeys, too. To reassign a key back to its original definition, just type control-shift-X, key, key. Watch out for the condition of the caps lock key. See the Key Redefinition section for more details.

MISCELLANEOUS SKEYS FUNCTIONS -- You can change your background attributes (**control-shift-**), change your cursor type (**control-shift-TAB**), dump the system time (**control-shift-7**) or date (**control-shift-8**) into your application, clear your screen (**control-shift-Z**), toggle keyclick (**control-shift-K**), program a quick function key (**control-shift-U**), replay the quick function key you just programmed (**control-shift-J**), or suppress some of SKEYS' displays when you are using SKEYS in a "programmed" mode (**control-shift-N**). See the Misc. SKEYS Function section for details.

SKEYS WITHIN SKEYS -- You can call up some SKEYS functions within other SKEYS functions. For example, at almost any time, you can print your screen, even while you are looking at the help screen or the ASCII chart. A chart is given in the SKEYS Within SKEYS section.

PROGRAMMABILITY -- In most cases, a SKEYS function can be programmed on a skey or regular function key. This allows one to automate many repetitive tasks. Sales tax can be automatically added to a calculator total, for example, with a single keystroke. Or you can set your system date and time without the DATE program. It is even possible to obtain square roots and even trigonometric calculations through programmability. Examples are given in the Application Notes appendix. There is even a SKEYSKEY program file that allows you to program your WordStar to make use of its memory-mapped capabilities -- with only two keystrokes!

SUPPORT PROGRAMS -- There are many programs provided to demonstrate the flexibility and capabilities of SKEYS. SKUTIL is the main support program. With it, you can save your redefined keyboards, your skeys, and your main and extra regular function key sets to program files that can be executed directly. This means you can quickly change your entire environment by merely running a short program.

Also included are some of the programs generated by SKUTIL, such as the Dvorak keyboard programs and the memory-mapped WordStar patch program. There are programs (FKEY and SKEY) which will automatically run a specified function key or skey, so that you can execute a function key as part of your "boot chain" -- when you boot, you wouldn't have to press a desired function key, since it could be automatically loaded for you.

SKEYS FUNCTIONS

QUICK ARROW KEY REDEFINITION

If you frequently run several different programs in succession, you may find that your backspace key is the wrong key for a particular application. This generally happens after you exit WordStar and run SuperCalc, or try to use the CP/M Plus operating system to copy files. Or, you may find that your right-arrow key does something strange while editing in WordStar.

Previously, you either limped along without the arrow keys, or you exited from your application to run SETUP or re-boot with an appropriate diskette -- one with a set of arrow key definitions compatible with your application.

Now, you can redefine your keys to be those for WordStar, or CP/M, or our own Special combination INSTANTLY, with a single control-shift keystroke, when you discover that you have the wrong arrow key definitions for the job.

To get:	Press:
CP/M arrows	**Control-shift-1**
WordStar arrows	**Control-shift-2**
The Special arrows	**Control-shift-3**

When you press one of the above, you will hear the characteristic "toodle" beep of SKEYS, telling you that something has happened. You can prove to yourself that the arrows are redefined by using them in your application; or you can look at them with the control-shift-F function. (Note: You cannot swap arrow key definitions with the above keystrokes while inside the control-shift-F function-key examine/edit mode.)

The Special arrow key definitions are a combination of the CP/M left-arrow key definition and the WordStar up-, right-, and down-arrow key definitions. We have found that this combination works well with 98 percent of the programs that run on the Executive, and suggest that you try it. When you run across one of the two-percent programs, you can easily change the arrow keys as described above.

If, for any reason, there is no room for the control-shift-1, -2, or -3 keys to swap out the arrow-key definitions, you will hear a foghorn "daaaaaah-duuuuuh" sound. This can only occur if you have actually used up all possible character spaces, AND you have actually removed a character assigned to an arrow key. No harm is done, but you will have to use control-shift-F and edit your arrow keys again to make room for redefining your arrows with control-shift-1, -2, or -3.

THE ON-LINE ASCII CHART

Control-shift-A brings up an ASCII (American Standard Code for Information Interchange) table, a chart showing the characters and their corresponding decimal and hexadecimal equivalents. Control characters will be displayed as underlined letters. To exit, just press the RETURN key. Your original screen will be restored.

An ASCII chart can prove useful in many ways. Certain printer initialization codes require a value (for instance, the number of lines in a page) that must be converted from decimal or hex into an ASCII character that the printer can accept and understand. When installing different programs, you might encounter the need to convert hex to decimal, or vice versa, and the chart gives the direct hex value for decimal numbers through 127.

One small use of the ASCII chart is to show the differences between what the computer thinks it is printing and what the printer actually prints. To see this demonstrated, bring up the ASCII chart with control-shift-A, then send it to the printer with control-shift-P. Compare the printout with the chart on the screen. You will note some differences, especially if you are using a daisy-wheel printer. (One note: Hex 7F, the ASCII delete character, will display as a half-intensity block, but will print as an asterisk. Most printers interpret 7F hex as "backspace and delete the last character", so it is not printed here.)

Those of you technically bent will find the chart convenient for purposes of cursor-addressing and the like.

NOTE: Every character on the chart can be generated directly from the keyboard, even if you don't see them printed on the keytops. The following characters are available on the standard Osborne-supplied keyboard:

control-comma -- left brace
control-period -- right brace
control-equals -- open-single-quote (accent grave)
control-left-arrow -- ASCII delete (hex 7F)
control-shift-caret -- control-caret (hex 1E, Exec home cursor)
control-shift-right bracket -- control-right-bracket (hex 1D)
control-shift-4 -- ASCII null (hex 0) [This ASCII character was originally on control-shift-2. It has been moved for incompatibility purposes, to accommodate our control-shift-1,2,3 arrow key redefinition keys.]

THE TEN-KEY BUSINESS CALCULATOR

There are two calculators available through SKEYS: A programmer's 16-bit calculator (discussed in a separate section) and a four-function calculator with memory. This calculator is often referred to as a business calculator to distinguish it from the programmer's hex calculator.

The calculator is accessed by pressing control-shift-C or control-shift-B. Your screen will appear to be cleared; it has been saved until you exit the calculator function. The screen is divided into three parts: The basic four-function calculator, a memory designated M, and a memory designated N. (At INOVA, the memories are called Memory and Nemory, naturally.) The cursor will appear near the lower left of the screen, in the basic calculator. A brief help message will appear on the bottom line. **To exit at any time, press the ESC key.**

This calculator is more than just a four-function calculator with memory. It is actually three calculators, a basic calculator and two memory calculators. The last answer calculated with the basic calculator is placed on the control-shift-= key, which means that you can recall the calculator answer into your application with control-shift-=. Note: Both the hexadecimal and business calculators store their last answer on control-shift-=. The last calculated answer determines what is "programmed" on that key.

Floating or Fixed Decimal Display

Internally, calculator results are evaluated to 16 binary-coded-decimal digits, and up to 14 digits are displayed in the "floating decimal" mode. However, you can select the number of digits to be displayed after the decimal point.

The calculator first assumes that you wish the answers displayed "fixed at two decimal places", which means that two digits will be displayed to the right of the decimal point, even if they are zeroes (like dollars and cents). The decimal point setting is displayed on the bottom line.

If you desire a different number of decimal places displayed, type the letter "F" followed by a digit from 0 through 9, or another "F". The number you select will be the number of digits to the right of the decimal; if you select "F" instead, the calculator will display all subsequent calculations with a "floating decimal point", where trailing zeroes after the decimal point are not displayed. (This makes the decimal point seem to move back and forth in the displayed numbers -- hence the term "floating".)

Regardless of the number of digits displayed, the calculator internally uses all 16 digits of the number in calculations. Displayed answers are rounded in the last displayed decimal place

according to the 5/4 round-up/round-down rule. So, if you divide 8 by 9 when you have the display "fixed" at 2 decimal places, you will see an answer of .89, but the calculator will use the number .8888888888888888 in subsequent calculations.

As an exercise, enter the calculator with control-shift-C, then press the following keys:

8 / 9 = F 2 + F 1 + F 0 + F F +

The plus key (+) merely redisplays the last number in the format defined by the F operation. (Constant math operations are supported, as are various ten-key-entry methods; they are discussed in "The Programmable Calculator" section and in the application notes section.) As you saw (or would have seen if you had "kept with the program") the displayed number varied considerably while the calculator ever so cleverly remembered the REAL value.

Arrow Keys Redefined

While the calculator "mode" is active, the up, right, and down arrows are redefined to be +, -, and *, respectively. This will allow you greater speed in keying in your problem. The +, -, and * in their regular positions, as well as the /, can also be used (if the corresponding keys have not been redefined with control-shift-X, or reassigned through skeys redefinition).

Digit and Operation Entry and Correction

To clear the basic calculator, press the letter **"C"**.

When you exit [by pressing the ESC key], if you wish to use the last answer in your application, DO NOT clear the calculator. **The last answer you calculated will always be placed on the control-shift-= key when you exit,** but clearing the calculator will store a zero on this key.

Digits and decimal point for a number can be keyed in directly, e.g., 1.2345. (That's a period at the end of the number -- we've found that most calculators allow only one decimal point per number...)

During number entry, the left arrow will delete the last digit entered. There is no clear-entry key; to clear an entry, just backspace to the beginning of the line. The entry of an operation (+, -, *, /, or a memory prefix -- M or N) will "enter" the number, preventing the use of the backspace (left-arrow) key to correct the number. Every operation entered will act like an equals (=) key, completing any pending operations previously entered. Both the RETURN and ENTER keys act just like the equals key.

The plus sign (+) signifies addition, telling the calculator to add the following entry to the previous entry or calculation; the hyphen or minus sign (-) signifies subtraction of the following

entry from the previous entry or calculation, the asterisk (*) indicates the multiplication of the previous entry or calculation by the following entry; while the slash (/) indicates the division of the previous entry or calculation by the following entry. Other signs, like ? for confusion, are fortunately not used here.

If there is no operation (+, -, *, /) pending, pressing the equals key after an entry will merely "enter" the number. If a calculation has just been completed, the equals key will do nothing. If an operation is pending, but no new entry has been made, the last operand will be echoed; this will have no effect on the pending operation.

If you accidentally enter a wrong operation, you can correct it by entering the right operation at the beginning of the line (backspacing if necessary), before you enter another number.

The operands (numbers) and the operations will be displayed in a paper-tape style, scrolling up from the bottom-left of the screen. Example: If you enter

F 2 4 + 5 / * 3 =

you will see:

```
 4.00 +                  (The "F" and "2" set the decimal)
 5.00 =
 9.00

 9.00 /
 9.00 *                  (The "/" was mistakenly entered)
 3.00 =
27.00
```

You might have entered

4 + 5 = / * 3 =

and accomplished the same result: The display would have looked exactly the same.

A new calculation can be started at any time after the equals key (=) is pressed. If an operation immediately follows the equals key, the previous answer is used as the new operand, as shown in the example above. (Some calculator manuals refer to this as "chaining".)

Change Sign

To change the sign of an entry, press the backslash key (\). This can be done at any time during number entry. If the number has already been entered, change the sign by multiplying it by a -1.

(He said, "Hi, you good-looking thing! What's your sign?"

She replied, "Negative.")

The Last Operand Key (and How To Square a Number)

If at any time you wish to recall the last operand or answer, press the "L" key. For example, if you wish to square the number 3.14159, you could do it this way:

3.14159 * L =

Hard Copy

You can take a printer "snapshot" of your screen at any time with control-shift-P, which will print the contents of the screen on your printer. The underlines on your screen will not print on your printer. If you wish, you can use control-shift-O with a special function key to label and time- and date-stamp your printout; see the Output-To-Printer section.

Memory Operations

All four functions can also be performed on the number in a memory (often referred to as the "contents of memory"), which will display in one of the columns to the right of the basic calculator. To perform an operation on memory, type "M" for memory M (or "N" for memory N), then type an operation key (again, +, -, *, or /). The memory calculations will look exactly like non-memory calculations. As with regular operations, all memory operations will complete any pending operation of the basic calculator and use the result in the memory calculation.

Upon entry into the calculator mode, the basic calculator and memories are NOT cleared. (This calculator has GOOD memories.) You can then use the memories to store constants. Here's the quickest way to store a number in a memory: Type the letter of the memory ("M" or "N") followed by "=" (equals). This stores the last entry or calculation in the basic calculator in the selected memory, after any pending operation is completed.

To enter a memory value in a regular (non-memory) calculation, press "M" or "N" followed by the letter "R" (for "recall"). To clear a memory, press "M" or "N" followed by "C".

Addition (+) and multiplication (*) aren't picky about the order in which the operands (numbers) are entered, but subtraction (-) and division (/) are. For those of you who know that a minuend is not a dance, please bear with us in this explanation.

In this calculator, when you perform a subtraction, the last number entered is subtracted from the earlier number or result, to give a new result. In division, the last number entered is divided into the earlier number or result. Examples (in floating-decimal mode):

```
                                        2 / 3 =
     2 - 3 =
```

will result in the following:

```
                                        2. /
     2. -                               3. =
     3. =
    -1.                  .66666666666667
```

In memory subtraction, the last entered number or calculated result is subtracted from the contents of memory to give a new result in memory. In division, the last entered number or calculated result (in the basic calculator) is divided into the contents of memory. For example, if the N memory contains 30 and you perform the following operation:

```
F 3 C 15 / 3 N /
```

the decimal will be "fixed" at 3 places, and you will see the following:

Basic Calculator	Memory N
0.000	30.00
15.000 /	
3.000 =	
5.000	
5.000 N/	30.000 /
	5.000 =
	6.000

To summarize: In memory arithmetic, the basic calculator value, whether a number just entered or calculated, will be added to, subtracted from, multiplied by, or divided into the number in a memory, and the result will be left in a memory.

Sum of Products

To perform "sum of products", use one of the memories to sum up the products you calculate with the basic calculator.

Example: Apples cost 59 cents a pound and oranges go for 69 cents each. You want 3 pounds of apples and 6 oranges. Although the calculator can't give them to you, it can tell you how much money you would need to buy them. You could enter:

control-shift-C	(to get into the calculator)
M C	(to clear the "M" memory)
.59 * 3 M +	(you'll see the answer displayed in the basic calculator and in memory M)
.69 * 6 M +	(the cost of six oranges is displayed in the basic calculator, while the cost of the apples and the oranges is in memory M)

If you wish to put the answer on the control-shift-= key, type:

M R =

Or, if there is a six percent tax on fruit:

M R * 1.06 =	(Recall sum and get the total with tax)

Now you can press the ESC key and "dump" the answer into your application (SuperCalc????) by pressing the control-shift-= key.

You see? With this calculator, it's easy to mix apples and oranges.

THE PROGRAMMABLE CALCULATOR

We think the calculator as described above is a fairly useful tool. With programmability, it becomes a powerful, flexible aid to your computing.

You can program a sequence on a function key or skey just as if you had entered it into your calculator. Then you can call it up with a single keystroke, and get your desired answer with just the press of a single key. You can have several of these keys programmed to perform different mathematical calculations, and to enter frequently-used constants into these calculations. You can even program a key to jump into the business calculator, start some calculations, and "prompt" you for an intermediate value (with control-shift-U, the "User-prompt").

Constant Calculations

The User-prompt key is actually a quickie function key. It is also the only function key you can program while right in the calculator. If you need to perform a constant calculation (where you will be adding, subtracting, multiplying, or dividing with a number that doesn't change), the User-prompt function key is the way to go.

As an example, if you need to frequently determine what the total cost of an item is, including the 25% state and local taxes:

Type **control-shift-U** while you are in the calculator. You will see the prompt "Enter data:" on the bottom line. Type in your constant calculation:

*** 1.25 =**

followed by RETURN to complete and enter the definition. (Substitute your own tax value, for the 1.25.)

You will notice that the calculation was performed immediately after you pressed the RETURN key. You have just programmed a total-with-tax key. To repeat the calculation, just press **control-shift-J.** Every time you press control-shift-J, you will see the calculator multiply the last answer or entry by 1.25 (or your own tax value) and display the new answer.

If you wish, you can assign control-shift-J to another key, like the single-quote key, for convenience:

control-shift-X, single-quote, control-shift-J

This makes the single-quote key (') your constant-calculation key. If you use control-shift-U to enter a new constant calculation, you can still use the single-quote key to repeat the new calculation at any time when you are in the calculator, until you turn off the computer or redefine the key.

Now you can enter the calculator with control-shift-C, and enter or calculate some number, say, the cost of four cans of oil at $1.09 each, and finally compute the total cost, with tax, with just one key:

```
4 * 1.09 '
```

You will see the following display (the decimal point is fixed at 2 places):

```
4.00 *
1.09 =
4.36

4.36 *
1.25 =                  (or whatever your tax rate is)
5.45
```

Nifty, no?

You can even program a function key or skey to jump into the calculator, prompt you for the total, calculate the tax, exit the calculator, and dump the answer into your application where you were when you hit the key you defined. Just program a key with almost the same keystrokes you would be using manually:

```
control-shift-N          (turn off console)
control-shift-C          (to enter the calculator)
control-shift-U          (to get user input)
* 1.25 =                 (the calculation)
ESC                      (to get out of the calculator)
control-shift-=          (to replay answer)
```

If those keys were programmed onto the open-square-bracket ([) key (see the Skeys section on how to do this), then, in the middle of a WordStar document, we could hit the open-square-bracket key and we'd see "Enter data:", where we could type in the total (or even a calculation, like 4 * 1.09), press RETURN (or ENTER), and watch as the total with tax was dumped right into our document, where our cursor was positioned.

The two extra functions, control-shift-N and -U, help the automation of the calculation: Control-shift-N will "hide" the calculator display, to avoid confusing the user with unnecessary clutter, and control-shift-U pauses the function key to ask for user input (here, that's the number to be totalled with tax). Please note that in this case, control-shift-U is used for a different purpose than in the previous illustration.

With the skeys programmability, you can set up a keyboard that will put functions like reciprocal and per cent calculations on a single key. One suggestion is to reprogram symbol keys (like the backslash, the single-quote, the open-square-bracket, etc.) with the CAPS LOCK (or ALPHA LOCK) key down, or ON. Then, when you

want to do certain calculations, locking the CAPS LOCK on will "convert" your keyboard. Don't forget to save any skeys definitions that you like with the SKUTIL program that we have provided.

Since you can program a calculator function on a function key, you could develop a square root function quite easily. Since a function key can call itself, iterative calculations can also be programmed without much difficulty. You could conceivably generate trigonometric functions through Taylor series expansions! (This is deliberately left as an exercise for the kind of reader who would!*)

* Doggone! Would you believe that we have one of "those" working at INOVA? The function key definitions for obtaining square roots as well as various trigonometric functions are given in the Application Notes.

GET STRING FROM VIDEO DISPLAY

With the **control-shift-G** (or control-shift-D) command, you can get (capture) all or part of a line of characters on your display for later recall.

To capture a string, type control-shift-G. The bottom three lines will change into a brief help message telling you how to use this function. Press RETURN to start the function; then use the arrow keys to move the block cursor to the start of the string that you wish to capture. (You will hear a series of quick tones as you move the cursor on the screen. This will always be an indicator of the control-shift-G "get video" mode.)

When you have positioned the cursor at the beginning of the string you wish to capture (it does not have to be the beginning of the line), press control-RETURN. This "captures" the characters from the cursor to the end of the line. Trailing spaces will be ignored. The display will return to normal, since the control-shift-G function was completed when you pressed control-RETURN.

Now, **to replay your captured video string, just press control-shift-V** (for video string recall). The string will be replayed back as if you typed it in anew. You can replay it back as many times as desired, by pressing control-shift-V as often as needed. (This is a quick way to duplicate a line in WordStar, or to re-enter a command under CP/M Plus when control-W is inappropriate.)

Please note: Every time you use the control-shift-G, control-RETURN combination, you redefine what is stored on control-shift-V. If you get into the control-shift-G mode and you do not wish to capture anything, press ESC.

There are some editing functions you can use in this mode to make moving about the screen quicker:

control-shift-A moves the cursor to the left margin of the screen, as does RETURN
control-shift-F moves the cursor to the right margin

and, since control-shift-P works while in this mode (meaning that you can print your screen at any time), you also have the ability to "pretty print" your screen by editing it with this mode. You can enter characters and spaces to change a screen's appearance before you print -- like removing unwanted information, and putting a label on top of the screen. Keep in mind that **if you want to print your screen, you must do so before you press ESC or control-RETURN.** Either of these exits will restore the original appearance of the screen before the editing was performed.

EXAMINE/EDIT/REDEFINE YOUR FUNCTION KEYS

One of the handiest features of SKEYS is the ability to look at and/or change your function keys. Just type **control-shift-F** and a window will open up on the bottom two-thirds of your screen, showing you your current set of function keys. You can examine them (press ESC to exit) or you can edit or totally redefine them -- in the middle of your program!

The arrow keys are indicated by U, R, D, and L, for up, right, down, and left, respectively. If you are confused by the control characters used here, instead of terms like WordStar or CP/M, don't worry -- control-shift-1, -2, and -3 will select CP/M, WordStar, or Special arrow key definitions when you are in the middle of an application, without having to enter this function-key redefinition mode.

You can select a key to edit in two different ways: You can specify the KEY NAME, like the number 5 or the letter U; or you can move the cursor with the up/down-arrow keys to position it in front of the definition you wish to change, then press RETURN or the right-arrow key to begin editing that key. The editing commands given in the NOTES section apply here, with the addition that **control-RETURN** saves the definition while **control-ESC** aborts and exits without changes.

You will note that while you are editing a key SKEYS will keep track of how many character spaces remain to be programmed, and display this count in the lower right corner. If you aren't watching and run out of character spaces, you'll hear SKEYS' "toodle" beep.

Some very practical enhancements are now available to help you get more out of your use of these keys: The function keys can call other function keys (or call themselves, up to 70 times); and now you can put labels or comments inside of a function key for your viewing pleasure only -- the comments will not be passed to your program when you press a function key.

Picture the following scenario:

You begin your WordStar(tm) edit session by programming the file name on a function key, say, #4. Now, you press control-O, which has the following string programmed on it:

```
WS 4.DOCM
```

Automatically, you will see on the CP/M Plus command line:

```
WS FILENAME.DOC
```

and WordStar will load and look for FILENAME.DOC on the logged drive. When you have finished editing, you can press control-7, which has:

KDP4.DOC[

This will save your file, and print FILENAME.DOC for you, again automatically. This is similar to the control-R function of WordStar, but here the filename on control-4 will always be used, and you can change it at will.

And then, the sequence

Y4.BAKM

will erase the backup file for you.

If you are a programmer, this becomes a very nice feature; you could conceivably have a command line to edit, assemble, and link a program, like so:

vdo 4.mac!m80 =4!l80 4/n,/p:100,4/eM

(Each exclamation mark tells CP/M Plus that a new command follows.) All you have to do is put the filename on function key #4 prior to executing this function key.

One function key can call another, which can call a third, etc., up to 70 levels of function key calling (Soooooweeee!).

If you happen to have a function key that refers to itself, it will be expanded 70 times. For example, if function key #4 has

Right now!4

then the phrase "Right now!" will appear to nag you 70 consecutive times in your text if you are editing a file with WordStar when you press control-4.

If function key #4 said,

Hello! 4 Goodbye!

it would be expanded like this (actual unretouched expansion):

```
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello!  Hello!  Hello!  Hello! Hello! Hello! Hello! Hello! Hello!
Hello! Hello! Hello! Hello! Hello! Hello! Hello! Hello!  Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
```

```
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye!  Goodbye!  Goodbye!  Goodbye! Goodbye! Goodbye! Goodbye!
Goodbye! Goodbye! Goodbye! Goodbye! Goodbye! Goodbye! Goodbye!
```

(Whew! WordStar acted like it was going to barf, for a while there! Lots of beeps, but it cleaned up its act when the function key finished. If you're the type that counted the Hellos and Goodbyes -- and noted 71 of each -- don't forget that there was already one Hello before the first of 70 "calls" to itself was made.)

(Somehow, I can't see a function key that selfishly calls itself as being all that useful, except to reformat a lot of diskettes or something. You might come up with something, like a destructive diskette tester, for example.)

You can also replay any keystrokes you captured earlier (with the control-shift-I interactive function key) into a function key. This is discussed in more detail in the description of the interactive function key.

COMMENTS IN FUNCTION KEYS

You can now put labels or comments in your function key definitions. For those control sequences for WordStar, or for initializing your printer, this can be very convenient. Now you can always know what each sequence does. **To put a comment in a function key, type a control-shift-", followed by your label or comment, followed by control-shift-" again to end it.** Your normal function key definition can follow, lead, or get out of the way -- it won't be affected when it comes time for the function key to be expanded. For example, here's a commented function key to center a line in WordStar:

```
* Center line *QC
```

This is the way it would look if you printed your screen while the function keys were displayed. On your screen the control-shift-" key will show up as a circle, and the Q would be underlined to indicate that it is a control character.

THE HEXADECIMAL/BINARY/OCTAL/DECIMAL CALCULATOR

This calculator is meant mainly for programmers. (If you need a "four-banger" with memory, refer to the ten-key business calculator section.) This "Hex" calculator, called up with **control-shift-H**, allows one to add, subtract, multiply, and divide 16-bit unsigned integers (range: 0-0FFFFh, 0-65535 decimal, etc.). Entry of numbers can be in one of four bases: Hexadecimal, binary, octal, and decimal. Answers can be recalled for next calculation, or even replayed (in one of four different bases) into your application or into one of the other SKEYS functions.

Number and operation entry is on a single line, from left to right. Pressing the RETURN key or the equals (=) key evaluates the expression from left to right. If no entry has been made, pressing the RETURN key or the equals key will re-enter the last result calculated. To recall the last result into your current calculation, type the letter "L" (for "last").

Numbers entered are assumed to be in decimal notation unless you end the number with a letter: H for hexadecimal, O for octal, or B for binary.

100	is	100 decimal
100h	is	100 hexadecimal, or 256 decimal
100b	is	100 binary, or 4 decimal
100o	is	100 octal, or 64 decimal

As you can see, the case does not matter. An entry must use numerals appropriate for the base -- e.g., DEADh and BADh are good and proper, but BADo or BADd or BADb are really bad. The SKEYS programmer's calculator will beep at you when you attempt to do illegal number conversions.

You can enter up to 79 characters of numbers and operations (+, -, *, and /) on a line, before you press the RETURN key or the equals key. You can mix cases; you can add, subtract, multiply, and divide numbers of different bases. Answers are subsequently displayed in decimal, hexadecimal, octal and binary (and even standard ASCII).

For your convenience, the up-, right-, and down-arrow keys are temporarily redefined as +, -, and *, respectively, while you are in the calculator mode.

You can also use the calculator to convert between bases. Just enter the number (with the appropriate suffix character to denote a base other than decimal) and press the RETURN key or the equals key. (Here, as elsewhere, the ENTER key is the same thing as the RETURN key.)

The answer is placed on the control-shift-= key. This answer can be used as well as the "L" key while you are performing calculations. Please note that the appropriate base letter is appended

to the answer while in the calculator mode ONLY.

Outside of the calculator, the answer on the control-shift-= key will be in decimal UNLESS you exit with control-H for hexadecimal, control-B for binary, or control-O for octal. As before, there will be no letter suffix indicating the base of the control-shift-= answer.

The answers can then be "played back" directly into the address of the memory editor in hexadecimal, for example, by exiting the calculator with control-H, entering the memory editor with control-shift-M, pressing control-A to set the address, then pressing control-shift-= to dump the hexadecimal answer into the address.

If an OVERFLOW occurs while using the hex calculator, such as when you add 0F000h to 0F000h, or multiply 4096 by 17, the calculator will beep and print "OVERFLOW", but the answer will contain the least significant 16 bits of the calculation.

NOTE: Since we cannot control data entry or manipulation, we cannot be responsible for wrong answers, or for wrong interpretation of right answers. Correct interpretation of answers are left as an exercise for the reader.

Examples of calculations:

F000h - 100h =

results in

DEC:**61184** HEX:**EF00** OCT:**167400** BIN:**1110111100000000** ASCII:..

(Boldface indicates full-intensity while rest of answer is half-intensity.)

How many 256-byte pages of memory does an 18K file occupy?

Since 18K probably means 18 1K-blocks, this translates to:

18 * 1024 / 256 =

which gives an answer of 72 decimal, 48 hexadecimal.

A reminder: The answer is always an integer. If division is performed, any remainder will be lost.

THE INTERACTIVE FUNCTION KEY

When SKEYS is loaded, you have the capability of creating a function key "on the fly" -- that is, you can save characters in a special function key buffer at the same time your program is acting on those characters.

In other words, SKEYS can remember keys you type as you type them, even though those keys are actually doing things (like logging a different drive, or doing a find/replace in WordStar). Then, at a later time, you can replay the keystrokes back, whenever you need to repeat the task, or even to save that remembered sequence on a function key or skey.

To tell SKEYS to remember subsequent keystrokes, type **control-shift-I** (as in Interactive). You will hear a quick series of four notes, played in rising order, from your computer. You won't see any evidence, but SKEYS is now recording all the keys you type that are passed to your application, until you type another control-shift-I or until you have typed 255 such characters (at that time, SKEYS will "close" the interactive function key automatically). In either case, you will hear a descending series of four notes, indicating that the interactive function key is "closed", and that no more keystrokes are being remembered.

To replay this interactive function key, type **control-shift-R** (for "replay"). SKEYS will then send all the characters in its interactive function key buffer to your application (CP/M, WordStar, etc.). If you had not "closed" the interactive function key, then it will be automatically closed before it is played back into your application.

Every time you press control-shift-I and hear the rising sequence of notes, you have told SKEYS to throw away any characters previously assigned to the interactive function key, and start recording anew. If there are no characters in the function key buffer, control-shift-R will do nothing. (If you hit control-shift-I twice in succession, opening and closing the interactive function key, there won't be any characters in the buffer.)

You may have noted that we said that the interactive function key will remember all the characters that are "passed to your application". This is because the interactive function key will not see any SKEYS keys, nor the keystrokes you press while you are in a SKEYS function.

As an example, suppose you had to write this manual, and found that you were typing the phrase "control-shift-" quite frequently. "Ah-hah!" you say, "I can put that on a function key!" You type control-shift-I just before you have to enter the phrase again, so that SKEYS will be remembering what you type. Then, for the last time (you hope), you type "control-shift-", followed by control-shift-I. You heard the appropriate rising and falling

notes, and now, every time you type control-shift-R, you see the phrase "control-shift-" dumped into your text. You can even enter the function key editor (control-shift-F) or the skeys editor (control-shift-S), and "dump" the phrase into a function key, by typing control-shift-R.

To carry the example one step further, suppose that you realized, in the middle of typing "control-shift-" after turning on the interactive function key (control-shift-I), that this would be a good thing to have on a permanent function key. First, you type control-shift-? to see what control-shift-key combination allows you to edit a function key. After finding it on the help screen, you press the RETURN key to return to your application.

You then type control-shift-F and select a function key on which to save the phrase (say, control-2). After you've begun to edit function key 2, you type control-shift-R to replay the phrase into the function key. Since you had not closed the interactive key before, you will hear it closed now, then you will see it "dumped" into the function key. Please note that the control-shift-? key, the RETURN key, the control-shift-F key and subsequent keystrokes in editing the function key were NOT played back into the function key.

One serious consideration of the replay of this function key: CP/M and some programs have this habit of checking for console input while they are performing a task. If they don't get a character that they expect (telling them to pause or turn on the printer, for example) they will throw those characters away.

This means that you can type (yes, we mean by hand):

```
DIR<cr>
```

and you might see a large collection of files displayed. Not immediately seeing the file you want (say, SKEYS.DOC), you might subsequently type a control-W followed by a space and the filename SKEYS.DOC, followed finally by a carriage return. The display would show:

```
DIR SKEYS.DOC
```

followed by an indication of whether SKEYS.DOC existed on that diskette.

BUT --

If you had saved all of the above on the interactive function key, then closed it, then tried to play it back after changing diskettes, you might see

```
DIR
```

followed by a directory listing, followed by nothing. This is due to that peculiarity of CP/M Plus that depends on the slowness

of human beings to work properly. You would have this same problem if you programmed a function key (through SKEYS, or without it through SETUP) and then executed that function key. If it has carriage returns in it, CP/M will practically throw away everything after the first carriage return.

Fortunately, WordStar and other programs generally remember every character you enter, whether by hand or through a function key.

The interactive function key contents can be played back into your application, or into the following SKEYS functions:

The function key editor (control-shift-F) -- edit mode
The skeys editor (control-shift-S) -- edit mode
The Get-Video-String mode (control-shift-G)
The typewriter mode (control-shift-T)
The memory editor (control-shift-M)
The hexadecimal calculator (control-shift-H)
The ten-key business calculator (control-shift-C)

You can also use control-shift-R to play back the contents of the interactive function key while the help screen or ASCII chart is showing. This is not very practical, however, as the first character cancels the SKEYS function, and subsequent characters are dumped willy-nilly into your application.

The control-shift-I open/close key can be programmed into a function key or skey. It will display as a blinking "I". When the function key or skey is played back, you will hear the rising/falling notes as appropriate, and the rules stated above for regular use of this function apply here in the same way. In this instance, however, if you end the function key with control-shift-I, you may need to "restart" your application with another key. (See the Notes section.)

The control-shift-R function cannot be programmed.

EXCHANGING FUNCTION KEY SETS

In addition to the main set of function keys in memory, there are four additional sets of function keys in SKEYS that can be loaded in two keystrokes, in the middle of any application program while it is waiting for a keystroke.

Simply type **control-shift-L** (for "load") followed by the number of the set that you want -- 1, 2, 3, or 4 (or M to re-load the main set). Each of these sets is EXACTLY like the main set in form and operation -- each can contain a maximum of 242 characters, each can be examined/edited with the control-shift-F function key editor, each can have function keys that call each other, etc., etc. The only requirement is that the set has to be loaded as the current set before any operations can be done to/with it.

When you first type control-shift-L, you will see the prompt:

Enter the set number to load (1-4) or M to load main set:

As soon as you type a number or the letter M, the current set is first saved in the computer's memory and the selected set is then loaded as the current set. As mentioned above, you can then examine and edit the new set with the control-shift-F command.

Each of these sets has its own space in memory, even the main set. This means that you can return to the main set at almost any time by simply pressing **control-shift-L** followed by **M**.

Saving These Sets

These sets are loaded into memory by special short programs created by the SKUTIL program supplied on the SKEYS distribution diskette. To save the current sets, you have to run the SKUTIL program, which is discussed in a separate section. You can have as many set-loader programs as you wish, and run them at the CP/M Plus prompt to load in a new group of four sets. (Note: The set-loader programs may also contain any skeys that had been defined when you ran the SKUTIL program.)

With Power Comes Responsibility

Since you have the capability of defining skeys (with control-shift-S) that can call the normal function keys, you must exercise caution if you also swap out function key sets. A reference to a function key by a skey will always apply to the current set loaded, and does not care what is actually programmed on the key. This provides great flexibility, but requires care in function key set design.

THE MEMORY EDITOR

Course description:

This course covers the use, misuse, and abuse of the SKEYS' built-in memory editor, both theory and application. The course is meant for the programmer and hacker, but hopefully provides enough material so that even the beginner can make use of this editor.

Prerequisites for this course:

CAUTION (and lots of it), and
Experience with DDT, SID, or other debugging tool, or
Six months of hacking, or
Four hours of EDFILE or SUPERZAP, or
Plain zeal and dogged determination

```
**********************************************************
*                                                        *
*    WARNING:  With this SKEYS memory editor,            *
*    anyone can modify a program or part of the          *
*    CP/M operating system in such a way as to           *
*    require a system re-boot to recover. If you         *
*    accidentally find yourself in this editor,          *
*    press ESC to exit.                                  *
*                                                        *
**********************************************************
```

The SKEYS memory editor is provided to give a user the ability to examine or change a few bytes in the Executive's memory. Access to the Executive ROM, the ROM's RAM, and video memory, as well as all applicable banks of the computer's regular memory, is provided. A memory-fill command and a block-move command are also provided. WordStar-like commands control the display scrolling, while arrow keys are used to move around within the displayed block of memory.

To invoke the memory editor of SKEYS, type control-shift-M. The bottom half of your screen will change to look similar to the following:

```
0100: C3 1A 04 00 00 00 C3 1B 01 C3 06 00 07 00 00 00      ¦................¦
0110: 4C 4F 41 44 45 52 20 20 FF 00 00 79 FE 3B C2 09      ¦LOADER  ...y.;..¦
0120: 01 C1 C5 21 00 00 39 31 BE 03 22 9A 03 C5 EB 22      ¦...!..91..".....¦
0130: 98 03 7C B5 F5 CC 00 02 F1 C4 30 02 D1 21 00 01      ¦..¦.......0..!..¦
0140: 7E FE C9 CA 9E 01 7A 3D B3 C2 5F 01 3A 0D 01 B7      ¦~.....z=.._.:...¦
0150: C2 5F 01 2A 0A 01 22 06 00 22 94 03 CD F8 01 2A      ¦._.*.."..".....*¦
0160: 9A 03 F9 AF 6F 67 C9 11 FE 00 2A 9A 03 F9 E1 E5      ¦....og....*.....¦
0170: 25 7C B5 EB 7D 44 C0 0E 09 11 53 03 CD 05 00 C3      ¦%¦..)D....S.....¦
BANK:01   START:0100 END:017F CURRENT:0100 BIN:11000011 DEC:195 OCT:303
TAB=switch side, ^S=switch banks, ^A=address change, ^E/^R=line/page up
^X/^C=line/page down,^B/^D/^O=binary/decimal/octal input,^F/^M=fill/move memory
^Q/^Z=cancel/save changes, ^ESC/^RET cancel/save changes and exit
```

Upon initial entry into the editor, the cursor will be positioned in the upper-left corner, after the address. A portion of memory is "displayed" in both hexadecimal and ASCII representations, with the ASCII displayed on the right. Some on-line help is given at the bottom of the screen.

Information appropriate to the block of memory and to the current cursor position within that block are shown below the hex and ASCII display and above the on-line help. This information includes the current bank, the start and end of the current block, the address of the current cursor position, and the binary, decimal, and octal representations of the byte displayed at the current cursor position.

You can enter hexadecimal values directly into the hexadecimal display, and ASCII values directly into the ASCII display. To toggle between displays, press the TAB key, or control-I.

While the cursor is positioned in the hex display, only memory editor commands, valid hexadecimal characters, the carriage return key (or ENTER key), and the arrow keys (for cursor movement) will be accepted as input. While the cursor is in the ASCII display, only editor commands, alphanumeric characters, or arrow keys will be accepted as input.

All entries are NOT actually saved to the displayed memory block until a control-Z or a control-RETURN is typed; any scrolling commands, like control-X, -C, -E, or -R, will ignore any changes not saved and re-print the screen anew.

ESC pressed at any level will abort the operation in progress and put you at the previous level. So, an escape pressed while setting an address will return the cursor to the previous position on the display, and ESC pressed again will exit back to your application program.

The Control-S Command

The current bank number is displayed near the lower left. To **switch banks, type a control-S.** The cursor will be positioned at the first digit of the two-digit bank number. Enter the bank number in two digits. As soon as the second valid digit is entered, the new bank will be displayed. Any entry referring to a non-existent bank may display all FFs, or a mirror-image of an actual bank; any changes to a mirror-bank will likewise be performed on the actual bank. (In a standard Executive, there are only two 64K memory banks, 0 and 1.)

You can also switch in (or out) the ROM and video banks by pressing **R** and/or **V** after control-S, but before you have typed the second digit of the regular bank to be displayed. If you do not wish to change the bank displayed, hit RETURN after toggling in/out the ROM or video banks. (NOTE: The ROM and video banks overlay only a portion of the RAM bank selected; see your Executive manuals for your memory arrangement.)

Pressing **C** will restore the bank select to bank 01 and clear the R and V toggles.

Please note that the ROM and video banks are NOT banks 7 and 8 in this bank-select scheme. This is to support the memory extension capability of the Executive. Also, you do not have to figure out "bank bits" to determine how to display memory in bank 5 -- just type in **05**. Valid bank numbers are 00 through 63 (there are 64 possible banks) in combination with the ROM and video banks.

The Control-A and Control-Shift-W Commands

Use control-A to set the starting address of the block of memory to be displayed/edited. Enter any valid hexadecimal numbers for the address, or replay an appropriate function key into the address, such as control-shift-= after a hex-calculator computation.

The **control-shift-W** command will replay the address AT THE CURRENT CURSOR POSITION of the memory editor into the starting address positions. This means that the memory editor can read an indirect address and use it to set a new starting address. For example, finding the current location of BDOS is easy, but you can have the computer fetch the current address and set it by itself. So that you can see how it's done, type in the following while in the memory editor:

control-A 0006 control-A control-shift-W

The first control-A prompts for the address, which you gave it: 0006. This displayed the block of memory starting at 0006 and positioned the cursor to the first displayed memory location. The starting location of BDOS is given here, low-order byte first. Then, the second control-A prompted for another address, which was given to it by control-shift-W, high-order byte first. This became the new starting address and a new block of memory was displayed.

(Incidentally, the control-shift-W command works just about everywhere else, too. It will always use the word at the cursor in the memory editor, even if you are in the hex calculator, or elsewhere. The value replayed will always be a word in hexadecimal.)

The sequence of keystrokes can be programmed on a function key, and the function key can be played back in the memory editor to repeat the command.

In addition to hexadecimal and ASCII entry, you can also enter a byte in binary, decimal, or octal. Just press **control-B** for binary, **control-D** for decimal, and (of course) **control-O** for octal, and your cursor will be positioned in the appropriate display, where you can make your entry in the selected base. When you type in the last character of the number, the entry will

be accepted and displayed in the hex and ASCII displays, but it will NOT have been actually saved.

Scrolling

You can "page" up or down, or scroll up or down a line, with **control-R for page up, control-C for page down, control-E for line up, and control-X for line down.** These commands change the starting address by either +/- 128 bytes (the page commands) or +/- 16 bytes (the line commands) and redisplay the contents of memory at the new starting address. Again, any changes made during the previous screen are lost if you did not save them with control-Z, which (of course!) we will discuss next.

Control-Z and Control-Q Commands

With **control-Z, you can save any changes** that you have made in the hex/ASCII displays.

> There are some areas of memory that you should not change -- the system interrupt stack or the SKEYS stack, for instance. Writing to the video display may be a fruitless task, as SKEYS will restore the previous video screen when you exit from SKEYS. (Patch the SKEYS video storage area, instead!)
>
> You can read the ROM, but writing to the ROM is actually writing to the font RAM, which can be interesting or irritating, depending on your mood.
>
> The control-Z command can also serve as a memory test of a byte -- the block of memory is written, then re-read and redisplayed. If a byte does not display what you wrote, it may be bad.

Control-Q will re-read and display the original block, cancelling any changes you made to the display but did not save with control-Z.

ESC and Control-RETURN

These keys are the counterparts to control-Q and -Z, adding one more feature after cancelling or saving -- **ESC** will cancel the editing and exit back to your application, while **control-RETURN** will save your changes and exit back to your application.

Fill and Move Memory

The **control-F** (for Fill) command will prompt you for the starting location, the ending location, and the byte to use in filling memory. If you type control-F, you will see the following in four steps:

```
Fill from 0000 up to and including 0000 with 00
                Confirm? (y/n) _
```

You will have to plug in the appropriate values, and some error checking is done to make sure the second address is the same as or above the first. (You can fill just one byte.) Enter all values in hexadecimal. When you press "Y" to confirm your entry, only the memory in the currently selected bank will be filled with the byte you selected.

To move memory, first figure out (possibly with the hex calculator) the number of bytes (in hex) of memory that you wish to move. Then type **control-M**. You will see:

Move from: **0000**

Put in the starting address of the block of memory you wish to move. After you press RETURN, you will see:

to: **0000**

Put in the starting address of the destination block of memory. When you press RETURN, you'll see:

for **0000** bytes

Now enter in the number of bytes that you want to move. This might be "dumped" into here from the hex calculator answer that was stored on control-shift-= (don't forget to exit the hex calculator with control-H for hex).

You will be asked to confirm this operation, as well, before the memory move takes place. You can only move memory within the current bank; however, if you are not using any function keys, you can use a portion of ROM's RAM to temporarily store 256 bytes of memory before switching to another bank. (These 256 bytes start at 2255 hex when the ROM is switched in.)

Notes on the Use of the Memory Editor

Combined with programmability, the memory editor can be a useful background tool for simple tasks, like setting the time and date without the DATE program, or selecting between different printers without the DEVICE program, in the middle of other programs. In these instances, a skey can be programmed to turn off SKEYS' output, prompt the user for a value, and perform the operation, without the user ever seeing the memory editor screen! See the Application Notes section for some examples.

PRINTER CONTROL

With the **control-shift-O** (for "output to printer") function of SKEYS, you can send to your printer a character, or a string of characters assigned to a regular function key, or a string of characters assigned to a skey.

The format is straightforward -- just type **control-shift-O**. You will see the following:

Press CHARACTER, FUNCTION KEY, or SKEY:

Now type a character, if you only want to send a character -- here, you could type a control-L if you want to send a formfeed to your printer while in the middle of a program. Or maybe you wish to change your dot-matrix printer output to compressed. You could then type a control-O for Epson compatibles, or control-shift-] for Okidata compatibles.

(Or maybe you don't want to send anything. Type **control-shift-ESC to cancel** this function.)

The character you type at the prompt above will then be sent to your printer, and you will hear the "toodle" beep to indicate that the character has been sent.

As the prompt indicates, you are not limited to one character. If you wish, you could send the contents of a function key or skey to the printer. In this instance, pressing a function key or skey will cause SKEYS to "expand" that function key or skey, but the characters will all be sent to your printer instead of your program. And, you are not limited to just a function key or skey -- you could type a control-shift-R in response to the prompt above, and send what you have recorded on the interactive function key (control-shift-I), or you could type a control-shift-V and send any string you captured with the Get Video String command (control-shift-G), or you could even send the last answer you calculated (control-shift-=) or the time and date (control-shift-7 and -8).

Since many of the above functions are programmable, you could select a function key that prints out a labelled screen-dump, complete with time- and date-stamping! It might look like this:

Screen Dump of **U**, taken on **8**, **7** JJJJ**P**

where the U, 8, 7 and P are blinking characters indicating control-shift-U (the User-prompt function), control-shift-8 and control-shift-7 (date and time) and control-shift-P (the screen-print function); the underlined J's are linefeeds to put an aesthetic gap between title and screen-print.

Then, every time you used the control-shift-O function with the above definition (assigned to a regular function key or to a

skey), you would get a formatted, titled, date- and time-stamped screen print. You could even include codes to emphasize your title, and compress your screen print, and restore the printer to normal after the print!

You might want to make this a shift- or control-key, to avoid too many accidental screen prints. One nice thing about the control-shift-U User-prompt function is the abort feature; you can type control-shift-ESC when you are asked to "Enter Data:", and control-shift-ESC again to cancel the rest of the function key. (See the "Miscellaneous SKEYS Functions" section for more information.) Since control-shift-ESC has no definition at your application, pressing it several times in quick succession will get you back to your application without putting any weird codes into it.

The control-shift-O function could be used in conjunction with the control-shift-L "load new function key set" function. You could have a complete set of function keys (say, set #4) defined to be nothing more than printer control strings, for setting form length, changing fonts, changing the number of lines per inch, etc. With the ability to put comments in your function keys, you could quickly look at them to see which key to press to set your printer to "emphasized", or "near-letter-quality" -- and you wouldn't have to have a manual handy to see which escape sequence does what.

Here's an example of a function key defined to set an Epson-compatible printer to "emphasized" print:

```
* Emphasized *[E
```

The asterisks here represent the circles you would see on your screen. These circles denote control-shift-" comment start/stop. The comment here is viewable on the screen ONLY -- it will not be sent to your printer or program when you press the function key. Spaces are put INSIDE the comment for clarity -- spaces outside the comment may prevent the function key from working properly.

Picture a list of commented printer control sequences. You could swap function key sets with control-shift-L, display them with control-shift-F, see this list, note which key has the control sequence you want, hit the ESC key to get out of the function key mode, then -- control-shift-O followed by your desired function key will change your printer setting to the mode you want. And you've done all this in the middle of your program!

Don't forget to restore your function keys back to the ones you want. If you want to avoid the swapping of function keys, you could define several skeys to accomplish the same task. The skeys have the advantage of being able to swap the regular function keys (since control-shift-L is programmable in a skey). Therefore, you can program a skey to swap sets, call a regular function key (which you already know has the code for compressed print), and swap the sets back when done. Then, when you press

control-shift-O followed by this skey, the sets are automatically swapped, and only the appropriate codes for compressed print (or whatever) go to the printer.

These function keys and skeys can be saved in a separate program file with SKUTIL, which is (of course!) discussed in a separate section of this manual.

(This space unintentionally left blank.)

PRINT YOUR SCREEN

At just about any time that your computer is waiting for input, you can send a "copy" of your screen to your printer with the **control-shift-P** command. This is known as a screen print or screen dump.

When you press control-shift-P, a carriage return is first sent to your printer, followed by the first line of ASCII characters on your screen (all 80 characters), followed by a carriage return/line feed pair, then the next line is sent, and so on, until the 24th line is sent with one last carriage return and line feed.

Graphics characters are converted into asterisks (*), as is the ASCII delete character (7F hex) which shows on the screen as a half-intensity block. Inverse video, blinking, half-intensity, alternate font, and underline attributes are ignored.

Spaces are also sent to your printer. Some printers will move rapidly back and forth when printing a line of spaces; others will rat-tat-tat without printing anything; in either case your printer will not be harmed.

If you wish, you can edit your screen before you print it. See the control-shift-G "Get String from Video Display" command; you can put titles on your screen print or remove unwanted material. Another method of labelling your screen-print is through the control-shift-O "Output to Printer" command, since the print-screen command can also be put on a function key. This is described in more detail under the control-shift-O command description.

You can also print your screen while already in another SKEYS function. This means you can get a copy of the ASCII chart, your calculations in either calculator, the function keys and skeys (during display mode only), the memory editor, and even the typewriter mode display (although you'll just be repeating what has already been sent to your printer). The only difference is that you cannot use control-shift-G to edit a SKEYS function display.

THE Skeys OF SKEYS

You are already familiar with the regular "control-number" function keys of your Osborne Executive. You know that you can, through SETUP, define a set of function keys that can make your WordStar editing go much faster and easier. If you have looked at the "Examine/Edit/Re-define Your Function Keys" section, you know how to look at and modify your set of regular function keys, even when you are in the middle of a program.

There are 22 additional function keys that you can use. These function keys are just like the regular function keys -- they can be examined, they can be edited, they can be used just about anywhere. BUT -- each one of these "skeys" can be up to 255 characters long, and each skey can be assigned to just about any key on your keyboard.

Call up the skeys display by pressing **control-shift-S.** You will see the words KEY and DEFINITION on the top line, followed by the skeys assigned keys on the left side of the screen, and their corresponding definitions to the right of the assigned keys. At the bottom is a small help-line which says: **UP or DOWN to move cursor, RIGHT to edit def, LEFT to change key, ESC to exit.**

For now, let's program a skey. (We can discuss the potential later.) A skey is programmed just like a regular function key. If you have just pressed control-shift-S, you should see a block cursor sitting on the vertical line separating the skeys and their definitions. Press the right arrow to begin to enter or edit a skey. Now, type your name. If you make a mistake, use the left arrow to backspace, and the down arrow to delete the mistake(s).

When you have entered your name, press control-RETURN. This saves your name on the skey and puts the cursor back on the vertical line. If you did not want to save anything, you would have typed control-ESC (or control-shift-ESC), which would have put your cursor back on the vertical line, and restored any previous definition of the skey.

Well, you've defined the skey, but you have not yet assigned the skey to any key (as shown by the "NO KEY" on the left). To assign this skey to a key, press the left arrow key when the cursor is sitting on the vertical line. The cursor will move to the beginning of the line.

Now, you can type the key or key combination that you want the skey assigned to. For example, you can assign this skey to shift-TAB. Hold down the SHIFT key, and press the TAB key briefly. The cursor will jump back to the vertical line, and you will see "SHFT" followed by an underlined "I". Now, for the sake of this tutorial, press the ESC key to return to your application. Get to a "safe" point in your program where you don't mind dumping your name into the program (you might get unusual results

executing this skey at the WordStar "not editing" menu). A safe point might be the CP/M prompt, or a new WordStar document file that you have opened for the purpose of learning SKEYS.

Now, press the shift-TAB combination. You will see your name printed out as if you had put it on a function key and "replayed" the function key. Actually, you have!

Now press control-shift-S again. You will note that your cursor is on the vertical line, on the same line as the last skey you were editing. Press the left arrow key to change the key, then press **control-shift-ESC** to "unassign" the key. You will note that your cursor is now back on the vertical line, and it now says "NO KEY" where it had said "SHFT I" with the "I" underlined.

You have just unassigned the skey. You have not lost the definition of the skey, however. Press the left arrow again, and assign the skey to the control-= key by holding down the control key and pressing the equals key. The cursor goes back to the vertical line, and the skey is now assigned to "CTL =". Press ESC (to get back to WordStar or CP/M or whatever) and try the new control-= skey.

If you had depressed the caps (or alpha) lock key, then pressed the equals key, you would have seen "CAPS =" instead of "CTL =". You can assign a skey to any normal, control, shift, or caps-lock key that you wish, EVEN the arrow keys! (While editing a skey, the arrow keys will take on the meanings discussed in the NOTES section on editing.)

If you assign a key to a skey, you cannot enter that key in the skey definition. If you wish, use the key in the skey definition before you assign it to that key.

We suggest that you play around with assigning skeys, to get a feel for the way your redefined keyboard works. If you find that you need a key that you've assigned to a skey, then **control-shift-X, key, key** will restore the key to normal. The skey definition will still be there -- just unassigned.

If you wish to save your skeys definitions, you can do so with the SKUTIL program we have provided. The SKUTIL program allows you to create an executable file on diskette that can contain your skeys, your four extra sets of function keys, your redefined keyboard... so that when that program is loaded, your skeys would be instantly available.

The SKEYS program looks for a file called SKEYSKEY.COM on your diskette when it is first run. If it does not find it, it merely loads in the SKEYS overlay (SKEYS.OVL). If it finds SKEYSKEY, it then "runs" SKEYSKEY. This way, you can automatically have a complete set of skeys/function keys load as soon as you boot.

If you do not save your skeys with SKUTIL, they will disappear when you press RESET or turn off the machine.

Potential!

A skey can call a regular function key. If you had "Peter Piper picked" on control-1, and "peck of pickled peppers" on control-2, then you could program a skey with the following:

> control-1 a control-2. RETURN A control-2 control-1.
> RETURN If control-1 a control-2, RETURN Where's the
> control-2 control-1?

It would look like this on the skey:

> 1 a 2.MA 2 1.MIf 1 a 2,Mwhere's the 2 1?

If you had assigned the skey to shift-TAB, then when you pressed shift-TAB in the middle of a WordStar edit session, you would see (like those of us at INOVA saw) the following:

> Peter Piper picked a peck of pickled peppers.
> A peck of pickled peppers Peter Piper picked.
> If Peter Piper picked a peck of pickled peppers,
> where's the peck of pickled peppers Peter Piper picked?

(Of course, if you "executed" this skey during a WordStar edit session, you would first see a lot of exclamation marks and hear a lot of beeps. Don't worry. WordStar is just telling you that you are "typing" too fast -- even though it won't miss a single character of the skey. The display will clear up almost immediately when the function key ends.)

You can also program control-shift characters into a skey definition. Things like the current time and date (control-shift-7 and -8) are easily entered.

In other situations, control-shift-7 would display the current time (or just the time elapsed since you last "booted" the computer, if you had not programmed the time). But, when defining a function key or skey, pressing control-shift-7 would put a blinking "7" into the function key definition. Then, when you "played back" the function key, the ACTUAL time would be played back.

With a little knowledge, you can actually program a skey to prompt you for the time and date, and automatically set them without the DATE program; or you can program your SKEYS calculator to do square roots and trigonometric functions. See the Application Notes section for details.

THE TYPEWRITER EMULATION MODE

At any time in the middle of an application, you can enter the typewriter emulation mode, to write a note to yourself or someone else, or to address an envelope.

To call up the command, type **control-shift-T** or control-shift-E. Your screen will be cleared and a one-line help message will appear at the bottom of the screen, with the cursor on the left margin just above it.

Now you can enter and edit text just as if you were using a fancy one-line memory typewriter. Control characters will show up as underlined capital letters. What you type will not be printed until you hit the RETURN key (and if you don't have a printer, not even then!). You can use the editor as described in the NOTES section.

To enter a RETURN without sending the line to the printer, type a control-RETURN. **To exit from this mode, type control-ESC at any time.** Anything on the line when you exit will be discarded.

The lines will scroll up on the screen as you press RETURN. This allows you to line up those addresses on envelopes.

NOTE: This mode automatically indents a fixed number of spaces before printing. The program is initially set to indent eight spaces; this can be changed (as shown in the Application Notes) if desired. A quick way to change this is to make your TAB key a skey; then program the skey to be a series of spaces or back-spaces (if your printer will take backspaces). Then enter the typewriter mode and enter the TAB key at the beginning of each line.

Both SKEYS and regular control-number function keys are expanded in the typewriter mode. If a carriage return code (control-M) is programmed into a function key, the code will be entered on the typewriter line as if you had typed a control-RETURN -- you will see an underlined M in your text. The line(s) will not be sent until you press the RETURN key directly. If a function key is expanded in the typewriter mode when there are more characters on the function key than are available on the typewriter line, the excess characters will be discarded.

A carriage return is always sent as a carriage return/line feed pair. If you wish to to just send an ASCII carriage return character, use the control-shift-O "Output to Printer" function.

The screen-print function (control-shift-P) will also work in this mode.

KEY REDEFINITION

You can redefine your keys at just about any time with the **control-shift-X** (for exchange key definition) function. The key sequence to redefine a key is:

control-shift-X, key-to-be-redefined, new-key-definition

(To cancel an accidental control-shift-X, type control-shift-ESC at the "KEY TO REPLACE:" prompt. Or, you can define a key to be itself.)

A "key-to-be-redefined" can be any regular key, any control key, any shift key, and any "CAPS LOCK" or "ALPHA LOCK" key. Basically, only the control-shift keys cannot be redefined.

The original keyboard definitions apply to the "new-key-definition". This way you can never get lost as to which key is defined as what.

For example, for SuperCalc you want the double-quotes (") key to be a lower case key. But you don't want to lose your single-quote (') key. Through control-shift-X, you could reassign the double-quotes key to the single-quote key, and the single-quote key to control-single-quote. Here's how:

control-shift-X, single-quote key, shift-double-quotes key

makes your single-quote key into a double quote key. And:

control-shift-X, control-single-quote key, single-quote key

fixes it so that when you press control-single-quote, you get a single-quote. The reason we didn't put the single-quote key on the original shifted double-quote key is simply to accommodate our habit of shifting for double-quotes on occasion.

To restore a key back to its original definition, just type:

control-shift-X, key, key

Please be careful of whether the caps (or alpha) lock key is depressed. This is also the case when you KNOW you've redefined the key, but the definition is just NOT there! Check the caps-lock key first.

To save your definitions on either the system tracks (so that the new definitions will be there when you boot) or in an executable program file, use the SKUTIL program, discussed in a separate section.

MISCELLANEOUS SKEYS FUNCTIONS

Selecting Screen Background Attributes

You can modify the "background" attributes of your display with the **control-shift-backslash** (\) key combination. When you press control-shift-backslash, you will see:

Enter choice: (0123456789:;<=>?)

Pressing the ESC key will exit from this command without changes. If you choose one of the choices, your background will immediately change, according to the following chart:

0 -- Normal (steady, full-intensity)
1 -- Half-intensity
2 -- Blinking
3 -- Half-intensity, blinking
4 -- Inverse video
5 -- Inverse video, half-intensity
6 -- Inverse video, blinking
7 -- Inverse video, half-intensity, blinking
8 -- Underlined
9 -- Underlined, half-intensity
: -- Underlined, blinking
; -- Underlined, half-intensity, blinking
< -- Underlined, inverse video
= -- Underlined, inverse video, half-intensity
> -- Underlined, inverse video, blinking
? -- Underlined, inverse video, half-intensity, blinking

This function is available during some of the other SKEYS functions; see "SKEYS within SKEYS", below.

Change Cursor Type

With **control-shift-TAB**, you can select which type of cursor you want for an immediate, temporary change. (If you want a more permanent change, use the OCC-supplied SETUP program.) Pressing control-shift-TAB several times will take you through a series of different cursors, from invisible through blinking and solid block, blinking and solid underline, and back to invisible.

System Time and Date

Control-shift-7 will dump the time (HH:MM:SS 24-hour format), and **control-shift-8** (MM/DD/YY format) into your application, or to your printer with control-shift-O, or into a skey or function key. See the "SKEYS within SKEYS" section, next.

Clear-Screen

You can completely clear your screen with the **control-shift-Z** command. Just press control-shift-Z.

Even the CP/M prompt will disappear until you enter a carriage return. Use caution when using this feature. For example, if you use this function during WordStar or SuperCalc, it may be a while (during which time you are scrolling and executing commands) before your screen returns to normal. This is because both programs assume that the data that they wrote to the screen is still there.

The clear-screen feature is most often used after using the TYPE command to display a WordStar file (where the soft carriage returns show up as circles), or when you want to get the screen ready for a "print-screen". If you have pressed control-shift-Z at the CP/M prompt, you can press RETURN to get the prompt to reappear.

The User-Prompting Function

The fastest function key in the computer is the **control-shift-U** function key. When you type control-shift-U, you will (quite suddenly!) see the bottom line disappear and the phrase "Enter data:" will take its place. Now you can type in a name, a phrase, a calculation -- anything but a RETURN, a function key reference, or a control-shift-key (except control-shift-ESC to cancel). When you have entered what you wish, press the RETURN key. Your bottom line will be restored, and what you have typed at the prompt will be replayed immediately into your application, or into one of the other SKEYS functions.

Also, the name, phrase, or what-have-you that you "programmed" with control-shift-U can be replayed back at just about any time with the **control-shift-J** "Just replay User-function-key" function. Every time you press control-shift-J, you will repeat what what last programmed with the control-shift-U function.

Please note that this means only what was last programmed on the control-shift-U function will be available, and that every time you use the control-shift-U function, you reprogram what goes on the control-shift-J key. (We mention the obvious here because some function keys that you or others have programmed might prompt you through the control-shift-U function, destroying what you had put there previously.)

As we have noted in the note, the control-shift-U function can be programmed in a function key or skey definition. So can control-shift-J, meaning that you can program a function key that prompts you once, then uses control-shift-J to use what you entered through control-shift-U.

You can abort the control-shift-U command (and any function key that may have used it) with control-shift-ESC. If control-shift-

U was part of a function key when you aborted it, you will be left where the function key had taken you -- the part before the control-shift-U function will have already been executed. This may be minor (some text may have been dumped into your program, for example) or it may be major -- you may be left in the middle of a memory edit session or some other SKEYS function. In the latter case, control-shift-ESC is a guaranteed exit, although it may have to be pressed more than once to "surface" back at your application (some SKEYS functions can be invoked within other SKEYS functions, and some SKEYS functions have more than one level).

The No-Output Command

With **control-shift-N**, you can suppress SKEYS output. Since we felt that this is of no value in the non-programmed use of SKEYS, this function is available only as a programmable function. You can only program it on a skey or regular function key. This is why it's not listed as a function on the quick-reference guide.

You could use this function for aesthetic reasons (e.g., when prompting for the time input, you would not want the user to see the memory editor display), for time reasons (calculations that are "automated" will run up to four times faster when the output is suppressed), or for practical reasons (with appropriate use of this function on calculator function keys, you can emulate different styles of ten-key calculation methods). See the Application Notes for examples of all the above.

It will display as a blinking "N". It works as a toggle; the first occurrence turns the SKEYS output off, and the second turns it back on. Only SKEYS output is affected -- your application program will receive the characters it would have received if control-shift-N had not been used, and will display them as before. Also, control-shift-N is only in effect as long as a function key is being "expanded" -- that is, until there are no more keys in the function key to be fetched. The SKEYS output will be automatically turned back on when the function key is finished, regardless of whether or not you remembered to turn on the output with another control-shift-N.

A WARNING: Since a function can be cancelled (one level at a time) with control-shift-ESC, it is possible to cancel a SKEYS function at a time that the output is suppressed. You might still be in the memory editor, for example, when you cancel a user-prompt function. This will still leave you in the memory editor, but you may not know this if the output has been suppressed. If you are not sure, or if you start to get some strange reaction from your program after you used control-shift-ESC to cancel a user-prompt, press it several more times.

SKEYS WITHIN SKEYS

There are different levels of operations possible for SKEYS functions. A quick example: you can print your screen (with control-shift-P) even in the middle of just about any other SKEYS function, like one of the calculators. Here is a chart of applicability for SKEYS within SKEYS:

```
+-------- CONTROL-SHIFT-FUNCTION IN PROGRESS
!
!                    FUNCTION AVAILABILITY
!
v
     J   L   0   P   R   U   V   X   7   8   \   =

A   ----------------------Yes-----------------------------
C   ----------------------Yes-----------------------------
dF  N   N   N   Y   N   N   N   N   N   N   N   N
eF  P   N   N   P   Y   P   Y   N   P   P   P   P
G   ----------------------Yes-----------------------------
H   ----------------------Yes-----------------------------
I   ------------------- SEE NOTE 1 ----------------------
M   ----------------------Yes-----------------------------
O   Y   N   N   Y   Y   N   Y   N   Y   Y   N   Y
dS  N   N   N   Y   N   N   N   N   N   N   N   N
eS  P   P   N   P   Y   P   Y   N   P   P   P   P
T   ----------------------Yes-----------------------------
```

Notes:
- P -- programmable only
- Y -- the top function can be executed within the function listed on the side
- N -- the top function cannot be executed within the function listed on the side
- d -- display mode
- e -- edit mode

Note 1: After the interactive function key is activated, only the non-SKEYS keys are placed into the buffer. SKEYS keys are processed immediately, and neither your application nor the interactive-function-key buffer will see them.

If a particular control-shift-key is not listed on the side, the top functions do not apply to it -- i.e., J, K, L, P, R, U, V, X, Z, \, =, TAB, 1, 2, 3, 4, 7, 8.

SKUTIL

SKUTIL is INOVA's name for the SKEYS utility that allows you to save the skeys and alternate function key sets that you create through the control-shift-S and control-shift-L, -F functions, as well as any new keyboards that you make through the control-shift-X function.

To perform these tasks, just insert a diskette containing SKUTIL on it into a drive (say, for example, the B drive). Exit from any application (like WordStar or SuperCalc) to the CP/M prompt. Then type:

B:SKUTIL<cr>

You will be given a menu of four options. You can save your keyboard definition to the system tracks of the diskette in the drive you specify; you can save a function key set to the system tracks; you can save your four extra function key sets and/or your skeys definitions to an executable file; or, bluntly speaking, you can quit.

If you select the first option to save your keyboard definition, you will be asked which drive. Upon successful completion of this task, you will be returned to the original menu. If SKUTIL fails to save your keyboard definition, you (and not your next-of-kin) will be notified.

The second option is similar to the first, except that it is a function key set that will be saved to the system tracks of the diskette in the drive you specify. You can write either the main set or one of the four extra sets to the system tracks with this option.

Your selection of the third option will present you with a series of Y/N (yes/no) questions. Do you want to save your four extra function key sets to a file? Do you want to save your skeys to a file? And, if you say no to the second question, you will be asked if you want to save your keyboard definition to a file. (It is saved automatically if you save your skeys definitions.)

If your answer to one or more of these questions is Y (for Yes), you will be prompted for a drive and filename that will contain these keyboard and/or function key definitions that you wish to save. (If you want the program to be automatically loaded by SKEYS, save it under the name SKEYSKEY on your boot disk.) After you answer with a filename, the diskette in the selected drive will be searched for a file of the same name as your input filename. If it exists, you will be asked if you want to delete it. If it is read-only, you will be prompted again.

You will be prompted once more before the new skeys file is saved to diskette. SKUTIL will give you an opportunity to input a

description of up to 30 characters for the file. This description will be displayed when the new program is run. (It will also be displayed if you "type" the file the way you would "type" a document file to read it under CP/M Plus.) If you do not wish to have a sign-on message, just press the carriage return key.

When the task of saving the file is completed, you will see the original menu again.

(Again, if SKUTIL fails, you will be told. Generally, failure is not fatal -- it may just mean that your diskette is write-protected, or there is no room, or a bad sector was encountered, or even simply that there was no diskette in the drive. You will be given the opportunity to try again.)

Now, at any time, you can switch keyboards and/or function key and/or skeys definitions by executing the program you generated. Programs saved by SKUTIL will be between 1K to 7K bytes in length, depending on how many characters are assigned to the skeys definitions, and whether or not you save the extra four sets of function keys, etc.

One note: SKUTIL is smart enough to distinguish between the main set of function keys and the alternate sets 1 through 4. If one of the alternate sets is loaded when SKUTIL is run, it will swap the function key set out and put the main set back in, so that it can save the alternate sets. However, the individual executable files that SKUTIL creates are not so considerate -- they will assume that the current set you have loaded is the main set, and overlay all the alternate sets, if they contain any alternate sets.

The files that SKUTIL creates can be freely distributed among SKEYS owners. They are smart enough to know which version of SKEYS is running (bank 0 or bank 2, G2 Systems Design, etc.) and will install themselves accordingly, when run. The supplied WSPATCH.COM file (to modify WordStar 3.3 for the Executive's memory-mapped video display) is one such example.

OTHER SUPPORT PROGRAMS

COPYSKEY is provided to make duplicate copies of individual skeys into other skey locations. There are 22 skeys, and COPYSKEY allows you to copy any skey into the position of any other skey. For example, you could copy the skey in the fourth position into the fifth position. The syntax is:

COPYSKEY x y<cr>

where **x** is the number (from 1 to 22) of the skey you want copied, and **y** is the number of the destination skey position (also from 1 to 22).

FKEY and **SKEY** allow you to execute a function key or skey with a program instead of pressing a key. This may prove useful in boot chains and other automated processes. The syntax is:

FKEY m<cr> or SKEY n<cr>

where **m** is a number from 0 to 9 (corresponding to the regular function keys) and **n** is a number from 1 to 22 (corresponding to the skeys).

WSPATCH is a program to patch your WordStar to use memory-mapped video instead of the standard console output routines. The result is a considerably faster WordStar. The main limitations are that such a patched WordStar has to use an inverse-video block for its cursor, and inverse video for marking blocks and displaying help and status information.

To run the patch, first make sure that SKEYS is running, then type:

WSPATCH<cr>

and follow the instructions. You will need a copy of WordStar and a diskette with the Osborne-supplied SID.COM on it. After running the patch, press RESET and reboot your WordStar diskette to try out your new WordStar.

The Dvorak Keyboard Files

Included on your distribution diskette are three files labelled QWERTY.COM, DVORAK1.COM, AND DVORAK2.COM.

DVORAK1 will load the Dvorak keyboard with qwerty control keys,
DVORAK2 will load the Dvorak keyboard with Dvorak control keys,
and QWERTY will return the keyboard back to normal.

If you wish, you can save your preferred keyboard to the system tracks with the SKUTIL program.

Here's the DVORAK layout (the numbers are standard in these versions):

```
ESC 1   2   3   4   5   6   7   8   9   0   [   =   \

 TAB '   ,   .   p   y   f   g   c   r   l   /   -

      a   o   e   u   i   d   h   t   n   s

       ;   q   j   k   x   b   m   w   v   z
```

Shift keyboard:

```
ESC !   @   #   $   %   ^   &   *   (   )   ]   +   ¦

 TAB "   <   >   P   Y   F   G   C   R   L   ?   _

      A   O   E   U   I   D   H   T   N   S

       :   Q   J   K   X   B   M   W   V   Z
```

On the DVORAK1 keyboard, the control keys for WordStar, etc., have not been changed. The DVORAK2 control keys correspond to the table above. These keyboards are by no means cast in concrete -- you can program any key to suit yourself.

It is suggested that you program a skey to load the qwerty keyboard back in if you are just learning the Dvorak keyboard. Then when you execute that skey, you can recover the qwerty keyboard, as well as your sanity!

APPENDIX A

Restarting

WordStar is the program that most often requires restarting. The folks at MicroPro made this word-processing program with lots of smarts; it will constantly be checking the keyboard for example, when it is printing a file in case you wish to stop the printing for some reason.

Let's say that you call up the "business" calculator (with control-shift-C) while WordStar is printing. You will be able to perform some calculations, and then exit back to WordStar by pressing ESC.

At least, you'll think you exited back to WordStar. The WordStar menu will reappear, BUT -- your printing will not continue, even though the menu says "P - Stop Print".

WordStar will not do anything -- until another character is typed on the keyboard.

This is due to WordStar's "sampling" of the keyboard -- it will check to see if you want to stop printing, as we noted above. It goes something like this:

> WordStar goes print, print, print, print... "Hey, CP/M -- ya got a character for me?"

CP/M asks his errand-boy, the Executive BIOS, the same question. The BIOS looks out the window at the ROM on the corner, then shakes his head. So CP/M tells WordStar, "Nah, not at the moment. My User hasn't pressed any keys."

WordStar goes back to print, print, etc., then repeats the question, over and over again, until either WordStar finishes printing, or CP/M finally says, "Yeah, I got one."

If WordStar got a "yes" answer from CP/M before it finished printing, it drops everything it's doing at the moment, and says, "Well, CP/M, which one is that? I'm not budging until you give it to me -- the User may want me to stop printing."

CP/M says, "Okay, just a micro-sec." He turns to the skinny kid and says, "Go get it, BIOS." The kid scampers out the door to oblige.

On his way back from the corner ROM, our poor BIOS is held up by that ruffian, SKEYS.

"Hey, punk, what's that you're carrying?" SKEYS demands of the skinny kid. "Is that one of my characters?" If

it isn't, SKEYS will let the BIOS take the key back to CP/M, and subsequently, to WordStar.

But if it is, things don't go well for our poor BIOS. SKEYS takes the character away from the kid, and sends him back to the corner ROM for more. BIOS even sometimes has to wait at the ROM for a long time, while the User takes his/her time using a SKEYS function.

Finally, SKEYS says to the exhausted BIOS, "Okay, punk, I'm done. You can go take a character back to CP/M and WordStar." But this means the BIOS has to go back to the ROM and sit and wait for the User to type a character. If the User doesn't know, the poor BIOS will wait, and wait, and wait...

(The preceding story you have read is mostly true. Only the author's name has been omitted to protect the guilty.)

So, SKEYS looks at every character you type, and if one of them is a SKEYS character, it takes control of the keyboard and processes that key. When SKEYS is done, it then asks for another character to replace the SKEYS character it "trapped" as its own. That's because the application (here, WordStar) asked for a key, and SKEYS wants to oblige, seeing to it that what WordStar wants, WordStar gets.

BUT, WordStar only asked for a key because it "checked the keyboard" and determined that a key was waiting, RIGHT THEN. When SKEYS trapped the SKEYS key, WordStar was "stood up", in a manner of speaking.

Any key then typed on the keyboard will "restart" WordStar. Please note that the key will be processed by WordStar as a key you meant to type, and some appropriate action will be taken. A left-arrow key is usually a harmless key to hit.

(You can actually perform a SKEYS function in the middle of a WordStar file-save operation! This is because, as mentioned above, WordStar always checks the keyboard, even when it is saving a file. If you print your screen while WordStar is busy like this, press another key, like the left-arrow key, to allow WordStar some peace-of-mind, so it can finish saving the file. No harm will be done.)

APPENDIX B

Resistor Fix for Contrast Problem

As discussed in the NOTES section of this manual, there are two alternatives to the contrast problem. One requires a manual switching of the background; the other, a more permanent solution, is described here.

We recommend that the following substitutions be performed by a qualified technician.

To solve the contrast problem on the internal monitor, replace R18 (originally about 470 ohms) on the main board with an 820-ohm or 910-ohm quarter-watt resistor.

To solve the contrast problem on an external monitor, replace R15 (originally about 100 ohms) on the main board with a 270- to 470-ohm resistor (this may vary, depending on the characteristics of the external monitor).

APPENDIX C

APPLICATION NOTES

Different function keys definitions and hints for using SKEYS are contained in this section. Unless otherwise noted, **SPACES ARE GIVEN FOR CLARITY ONLY**. Other algorithms can be used, which may reduce function key overhead and increase speed. Any function given with only a single function key can be assigned to either a skey or a regular control-number function key. Functions given with two function keys need to have the second function key on a control-number key, but the first can be on either a skey or a regular function key.

Setting Date and Time:

control-shift-N control-shift-H 2769 + control-shift-U =
control-H control-shift-M control-A F8F4 control-shift-=
control-Z control-A RETURN control-shift-W control-shift-U
control-Z ESC

NOTE: **SPACES ARE GIVEN FOR CLARITY ONLY!**

When this function key is executed, the user will be prompted to enter data. Only the day of the month is entered here -- a number from 1 to 31. When the user presses the RETURN key, s/he will be prompted again to enter data. Now the time is entered here, in either four or six digits, in the format HHMM or HHMMSS, followed by the RETURN key. If just the RETURN key is pressed, the time will not be set.

***** CAUTION! *****

> **DO NOT ENTER MORE THAN SIX DIGITS FOR THE TIME.** You will crash your system and have to reboot if you enter more than six.

After the second RETURN, the prompt will disappear, and the date and time have been programmed, as shown when control-shift-7 and control-shift-8 are pressed.

Please note that the date is calculated in the hexadecimal calculator from the last date of the previous month (here, July 31, 1985) and therefore this function key must be updated once each month. There have been 2769 days from January 1, 1978, to July 31, 1985. Add the number of days in the month for every month that passes, i.e., the number for September would be 2769 + 31 days in August, or 2800; the number for October would be 2830, etc. Replace the 2769 figure above with the correct number for the current month.

The user will not see anything, since the control-shift-N "No Output" toggle has been used to keep the hexadecimal calculator

and the memory editor from showing.

Function Key to Select Parallel Printer (known good for BIOS versions 1.0 and 1.1):

control-shift-N control-shift-M control-A F8C740 control-Z ESC

Function Key to Select Serial Printer (same consideration as above):

control-shift-N control-shift-M control-A F8C710 control-Z ESC

These function keys may be used to choose between printers in the middle of a program, without having to use the DEVICE program. These functions can also re-assign a printer that has been "unassigned".

(Author's note: The second line was very similar to the first. While using WordStar to produce this document, the author used the control-shift-G function to capture the first line, so that it could be repeated for the second line. Before pressing control-RETURN, the author (that's me!) pressed control-shift-F, space, RETURN to get rid of that "<" symbol at the far right of the display. Then, after pressing control-RETURN, the author pressed control-shift-V in the appropriate spot to replay the line.)

Function Key to Calculate the Trigonometric Sine of a number in degrees (accuracy: -90 < angle < 90 degrees -- to 8 decimal places):

Function must be executed in calculator mode, or precautions taken to insure same. Argument (that's the user input) is assumed to be in display as last answer or entry.

control-shift-N * 3.1415926535898 / 180 = n = m = m * control-6 6 n - control-6 20 n + control-6 42 n -control-6 72 n + control-6 110 n - control-6 156 n + n r control-shift-N =

where control-6 contains: *** m r /**

This function key is actually two function keys. Control-6 is used here, but any convenient one can be used. The "sine" key can prompt for user input by using control-shift-U. When the first function key is programmed, it will look like this on the screen:

N*3.1415926535898/180=n=m=m*6n-620n+642n-672n+6110n-6156n+nrN=

where each capital N letter will be blinking.

When this key is executed, allow a few seconds for calculations to complete. To see just how much the calculator is doing, remove the control-shift-N keys from the function key, and re-execute the function.

The first part of the function converts degrees to radians, and the second part does the actual evaluation of the sine. This function key is based upon the Maclaurin series expansion (which is a Taylor series expansion evaluated at zero) of the sine function. You could get greater accuracy by expanding the function above with more terms; you can see that the terms are alternately added and subtracted, and the coefficients are simply 2*3, 4*5, 6*7, 8*9, 10*11, 12*13, etc. This is but one method of calculating the sine of a number.

P.S.: The sine of 30 degrees is supposed to be .5, exactly.

Calculating Square Roots:

Two function keys are recommended. The first looks like this:

m c n = + 1 / 2 m = control-3 control-3 control-3 control-3 control-3 control-3

and the second looks like this:

n r / m r + m r / 2 m =

Note: **SPACES ARE GIVEN FOR CLARITY ONLY!** (Sheesh!)

The same input requirements for the sine function, above, apply here. To increase accuracy for large numbers, increase the number of control-3 function key references.

Insert a control-shift-N at the very beginning of the first function key to hide calculations and increase speed.

Reciprocal Calculations:

This function takes the last answer or entry and calculates the reciprocal (1/x) of the input. Memory N is used.

control-shift-N n = 1 / n r control-shift-N =

Hint: You might want to reserve a memory (like memory N) for temporary calculations for all your "function" keys, to avoid conflicts.

A Ten-Key Calculator (one version):

On your skeys, put the following definitions:

KEY	DEFINITION	
CAPS =	Nm+N+Nn=cnr=N	<-------all capital N's are control-shift-N's
CAPS -	Nm-N-Nn=cnr=N	<-------same here
CAPS T	mrMNmcN	<-------underlined M is carriage return
CAPS C	cNmcN	<-------capital N's still control-shift-N's

(To program these functions, program the skey definitions first, then go back and assign the keys as shown. Otherwise, you will be hearing a "toodle" beep while programming the definitions when you press the equals key and the hyphen, since they were "assigned to skeys" already!)

Now, when you depress the CAPS LOCK (or ALPHA LOCK) so that it is down, your keyboard will be converted so that the hyphen key is SUBTRACT THIS ENTRY, the equals key (the unshifted plus key) is ADD THIS ENTRY, the T key is DISPLAY TOTAL, and the C key is CLEAR CALCULATOR.

To use this calculator, always press the CAPS C key (the C key when the CAPS LOCK is down) to clear it. The equals key here is really the ADD key. The hyphen is the SUBTRACT key.

Here's an example of this calculator in use (decimal point fixed at 2):

Press:	Display shows:	Remarks
C	0.00	Calculator cleared
45 +	45.00 +	equals key pressed
25.16 -	25.16 -	25.16 is subtracted
85.92 +	85.92 +	
T MR		T pressed, and...
	105.76	Total displayed
+	105.76 +	Using previous total in next calculation

and so on...

You could develop this type of calculator for your checkbook balancing and the like. The nice feature of this is that a screen print of this looks very much like a paper-tape calculator printout. We suggest experimenting with the function key definitions to obtain the "ten-key" simulation that you want.

WARNING: If you forget that the CAPS LOCK key is down, and you go on to other tasks (for example, editing in WordStar), you may

get some unusual responses when you press certain keys. If this is the case, press control-shift-ESC several times, then press control-shift-S and look at your skeys. You may find the source of the problem right there. If you need to have the CAPS LOCK down, just unassign any problem skey with control-shift-ESC, and continue processing in peace.

Changing the Typewriter Margin:

With the memory editor, you can quickly change the number of spaces that the typewriter indents before printing each line. First, you need to know which version of SKEYS you are using -- whether it is a bank 0 SKEYS or a bank 2 SKEYS. If you are not sure, SKEYS will tell you when it signs on.

Type **control-shift-M**. This will get you into the memory editor.

If you have a bank 0 version, type the following:

control-S 00 control-A 4005

If you have a bank 2 version, type the following:

control-S 02 control-A 5005

The memory editor's cursor will be on the first of two digits (09 when you first do this). Type in the desired starting column (from 1 to 254, in HEXADECIMAL) followed by

control-Z ESC

to save the new column and get out of the editor. If you need to change the margin again, just type **control-shift-M**. The bank number and address are retained from the previous use of the memory editor.

One convenient use of this is in addressing envelopes that already have a return address. You can set the left margin here to be 50 or so spaces, so that you don't have to type in 50 spaces on each line in the typewriter mode.

If you wish, you can patch a different value permanently into the SKEYS overlay, so that you don't have to continually patch it with the memory editor. Use EDFILE or SUPERZAP or SID to patch the sixth byte in SKEYS.OVL to the desired value. Make sure that the byte you are changing is originally a 9.

The SKEYS overlay will start at 4000h (bank 0) or 5000h (bank 2). At 4002h (or 5002h) is the zero-relative offset (from the beginning of the SKEYS overlay) to the SKEYS data section. At 4006h (5006h) is the ADDRESS of the "recursion maximum value" byte. The byte (<=70) indicates the maximum number of recursion levels.

APPENDIX D

SKEYS TECHNICAL INFORMATION

OPERATION

This section is meant for hackers only. It discloses details concerning SKEYS operation and memory usage.

SKEYS resides in either Bank 0 or Bank 2 of memory. If you own the INOVA 500 memory expansion card, you have a Bank 2 version. All other versions run in Bank 0. Bank 0 memory is reserved for SKEYS by the PATCHCPM utility by restricting the number of LRU data buffers allocated to CP/M Plus.

SKEYS occupies the memory listed below for the indicated version:

	Bank 0 SKEYS	Bank 2 SKEYS
Start	4000h	5000h
End	AE00h	C000h

SKEYS works by intercepting all calls to the RAM BIOS CONIN vector. The SKEYS loader program creates the front-end environment for SKEYS. The front-end routine sits in an unused area of the Device Table. (On most standard systems this area starts at FABAh.) The first eight device slots are untouched. The last slot is also left unmolested to provide compatibility with the public-domain utility EXCLOCK. The SKEYS front-end routine starts at DEVTBL + 64d (8 slots by 8 bytes each) + 3 for a special SKEYS Expansion Interface. (More on that later.) Therefore, the SKEYS loader program redirects the CONIN vector to DEVTBL + 67d.

At this location, the actual CONIN routine is called. Then the front-end routine saves the current bank configuration, selects the SKEYS bank and calls SKEYS. SKEYS processes the character input and then returns to the front-end which in turn returns to the caller of the CONIN routine. If SKEYS needs more characters during its processing, it calls the ROM. While SKEYS is enabled, only the ROM, VIDEO and SKEYS banks are enabled. Bank 1 (TPA) is completely shut out.

The SKEYS loader program is also responsible for patching the keyboard tables (which start at 2015h in the ROM's RAM) with the codes necessary to provide access to SKEYS functions. Primarily, it is the fifth table, the control-shift table, that is patched. Below is a list of the codes used for the various SKEYS functions:

Control-Shift Key	Code
J	DDh
W	DEh
8	DFh
7	E0h
U	E1h
N	E2h
Y	E3h
\	E4h
ESC	E5h
1	E6h
2	E7h
3	E8h
Q	E9h
C	EAh
I	EBh
?	ECh
=	EDh
H	EEh
L	EFh
K	F0h
Z	F1h
TAB	F2h
P	F3h
O	F4h
R	F5h
T	F6h
A	F7h
F	F8h
S	F9h
M	FAh
G	FBh
V	FCh
X	FDh

Through the assigment of SKEYs to various keys, the keyboard tables may be patched with other values. The codes used for the 22 SKEYs are C0h to D5h.

Whenever a SKEY or function key references another function key, the code used in the definition is defined as KEYTBL CODE + 10h. For example, if F1 (function key 1) references F3, the code placed into F1's definition is 93h. (The original code was 83h. 10h is added to identify the special reference.)

MEMORY LAYOUT

Contained in the first sector of SKEYS are several pointers and variables. This is a discussion of each and what it means. Below is the list to consider:

Offset from Start of SKEYS	Length	Purpose
02	2	Offset to SKEYS data block
04	1	Current length of SKEYs
05	1	Typewriter mode left margin
06	2	Address of maximum key recursion level value
08	2	Address of current key recursion level value
0ah	2	Address of Expansion Interface

Offset 02: This word points to the SKEYS data block. That area consists of the following elements:

DATA	SIZE
"Master" keyboard tables	0140h
Extra function key sets	0401h
SKEY definitions	1600h

Each SKEY contains a header indicating how many bytes of the SKEY are defined, followed by the definition.

Offset 04: This byte controls the maximum length of each SKEY. By changing this byte and loading a new set of SKEYs, you can effectively lessen the amount of memory reserved for the SKEY definitions and thereby reserve some memory for another purpose. All of the SKEY definition files created by SKUTIL and other support utilities respect this length indicator. (Try setting it to 15h and then running your favorite SKEYSKEY file!)

Offset 05: Change this byte to adjust the left margin used by the typewriter mode.

Offset 06: As the function keys call each other, or the SKEYs call the function keys, or the function keys call other SKEYS functions, a stack of pointers is developed to control the "unwinding" of each key as it finishes and returns to the level that invoked it. The maximum size of this stack is 70d levels. You may effect this "on-the-fly" by setting the byte at this address to the maximum level of recursion you can endure. However, that value cannot exceed 70d.

Offset 08: This address points to the current level of key recursion. When this reaches zero, function key translation will cease after the current key finishes.

Offset 0Ah: This is the address of the SKEYS Expansion Interface (SEI), which brings us to our next and final topic...

EXPANDABILITY

Whenever SKEYS encounters a control-shift-Q (E9h) the following steps occur:

1) The byte at SEI + 0 is checked. If zero, no action is taken and the E9h code is ignored. Otherwise, this number is considered to be a bank configuration to call.
2) The byte at SEI + 1 is loaded as the high order byte of the address to call in the specified bank.
3) The screen is saved and cleared. A common memory stack is established in the area of FFF1h-FFFCh. The address is called.
4) When the called routine returns, the screen is restored and control is returned to the calling program.

In other words, this is a user-hook provided so that other software can take advantage of SKEYS' ability to save the screen and monitor keystrokes. One example of the use of this hook is the Print Buffer software for the INOVA 500 memory expansion card.

If you establish your own software to make use of this facility, you must ensure that it makes NO use of the SKEYS stack. Your software MUST establish its own stack immediately.

UPDATES, LICENSES, WARRANTIES

Sending in the registration form will allow us to notify you in the event we discover a serious bug or problem in your version of the software.

As a registered owner, you will also be entitled to purchase any subsequent revision of this program for $10 for each product update, plus $3 shipping & handling (Texas residents add $0.80 sales tax). We reserve the right to change this policy.

Purchasing this program entitles you to a single purchaser license, which is permission to run the software on up to five computers at a time. Users with six or more machines may purchase additional licenses for 50% of the retail price for machines 6 through 10, 20% for machines 11 through 15, and 10% thereafter.

Multiple copies of the program and documentation can be purchased in a single order, with a single check for remittance. Copies 1 through 4 are at 100% of retail price, copies 5 through 9 are 90% of retail, copies 10 through 19 are 80%, copies 20 through 49 are 70%, and 50 or more are 60% of retail.

THIS SOFTWARE AND MANUAL ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THIS PROGRAM IS ASSUMED BY THE LICENSEE.

HOWEVER, TO THE ORIGINAL PURCHASER ONLY, INOVA WARRANTS THE MAGNETIC DISKETTE ON WHICH THE PROGRAM IS RECORDED TO BE FREE FROM DEFECTS IN MATERIALS AND WORKMANSHIP UNDER NORMAL USE FOR A PERIOD OF NINETY DAYS FROM THE DATE OF PURCHASE. IF DURING THIS NINETY-DAY PERIOD THE DISKETTE SHOULD BECOME DEFECTIVE, IT MAY BE RETURNED TO INOVA FOR A REPLACEMENT WITHOUT CHARGE.

YOUR SOLE AND EXCLUSIVE REMEDY IN THE EVENT OF A DEFECT IS EXPRESSLY LIMITED TO REPLACEMENT OF THE DISKETTE AS PROVIDED ABOVE. IF FAILURE OF A DISKETTE HAS RESULTED FROM ACCIDENT OR ABUSE INOVA SHALL HAVE NO RESPONSIBILITY TO REPLACE THE DISKETTE UNDER THE TERMS OF THIS LIMITED WARRANTY.

ANY IMPLIED WARRANTIES RELATING TO THE DISKETTE, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO A PERIOD OF NINETY DAYS FROM DATE OF PURCHASE. INOVA SHALL NOT BE LIABLE OR INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THIS PRODUCT. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

SKEYS 1.1 -- page 70

INDEX

alpha lock key 23,48,63
application notes 60-64
arrow keys .. 8,10,11,14,17,29
ASCII chart 8,11,15

background attributes 49
backslash 11,18,49
bank 2 5
bank, ROM 36,37,39
  video 36,37
base (number) 29
BIOS 5,57-58
boot chain 13,55
boot diskette 6,7

caps lock key 23,48,63
change sign 18
clear screen (ctrl-shft-Z) 50
comments in function keys 28
control keys
  -A (set starting addr) 37
  -B (select binary) 11,30,37
  -C (page down) 36,38
  -D (select decimal) 30,37
  -E (line up) 36,38
  -ESC (exit) ... 10,13,26,47
  -F (fill memory) 38
  -I (TAB, switch sides) 36
  -M (move memory) 39
  -O (select octal) 37
  -Q (re-display page) ... 38
  -R (page up) 36,38
  -RETURN (save changes
    and exit) .. 10,26,36,38
  -S (select bank) 36
  -X (line down) 36,38
  -Z (save changes) ... 36,38
control-shift keys 7
  -1 (set arrows CP/M) 11,14
  -2 (set arrows WS) .. 11,14
  -3 (special arrows) 11,14
  -4 (ASCII null/hex 0) .. 15
  -7 (system time) 13,49
  -8 (system date) 13,49
  -? (help screen) 7
  -A (in editing) 10,25
  -A (ASCII chart) 11,15
  -B (business calc) .. 11,16
  -backslash 13,49
  -C (ten-key calc) ... 11,16
  -D (grab display string) 25
  -E (typewriter emul.) 12,47
control-shift keys (cont.)
  -ESC (exit) 7,51
  -equals 11,16,17,29,30
  -F (function keys) .. 11,26
  -G (get video string) 11,25
  -H (hex calculator) 11,30
  -I (interactive fkey) 11,31
  -J (replay user fkey) 13,50
  -K (in editing) 10
  -K (toggle keyclick) 7,13
  -L (load alt fkeys) 12,34
  -M (memory editor) .. 12,35
  -N (no-output function) 51
  -O (output to prntr) 12,40
  -P (print screen) ... 12,43
  -R (replay i-key) 12,31-33
  -S (skeys) 12,44-46
  -T (typewriter) 12,47
  -TAB (change cursor) 13,49
  -U (user-prompting quick
    function key) 13,22,50
  -V (replay video str) 11,25
  -W (replay word at memory
    editor cursor) 7,37
  -X (exchange key def) 13,48
  -Z (zap screen) 13,50
COPYSKEY program 55
cursor control 13,49

DATE program 39
date, system 49,60
DEVICE program 39
Dvorak keyboard 56

editor, memory 12,35
editor, function keys line 10
exchanging fkey sets 34
EXECST.COM 6
exit via ctrl-shft-ESC .. 7,51

fastest function key 50
fixed decimal point 16
FKEY program 55
floating decimal point 16
function key(s)
  interactive 31
  exchanging sets of 34
  regular 11,12,26-28,34
  saving 34,53-54
  skeys 12,44-46
foghorn sound 14

get/grab video string .. 11,25

help screen 7

interactive function key 11,31

keyclick toggle 7
key redefinition 13,48
key redefinition, arrow 11,14
keyboard redefinition .. 13,48
keyboards, saving 53-54

L key 19,29
last operand/result 19,29
levels of SKEYS 51,52

margin, typewriter 47,64
MAKEST 6
memory, calculator .. 11,16,19
memory editor 12,35
memory-mapped WordStar 55

no-output command 23,51

overflow 30

paper-tape calculator 63
parallel printer select ... 61
PATCHCPM program 5
printer control 12,40,61
PROFILE.SUB 6
programmability 22,52
programmer's calculator ... 29

quick function key 13,50
QWERTY program 56

reciprocal calculations ... 62
regular function keys
..... 11,12,26-28,34
replay interactive function
key 12,31-33
replay captured video
string 11,25
resistor fix 59
restarting 9,57
ROM 36-37,39,57-58

saving fkeys, keyboards 53-54
serial printer select 61
SKEY program 55
skeys 8,12,44
SKEYS within SKEYS 52
SKEYSKEY.COM 45,53
SKUTIL program
..... 4,12,13,24,34,45,53
squaring a number 19
square roots 62
sum of products 21
system date 49,60
system time 49,60

time, system 49,60
typewriter emulation mode
..... 10,12,47
typewriter margin 47,64

Unassign skey 13,45
User-prompting function ... 50

video attributes 49
video bank 36,37
video font RAM 38
video string capture ... 11,25
video string replay 11,25

This SKEYS documentation was
written,
proofread,
and wept over
by
M. Joel Guerra
R.I.P.